

# **PENGEMBANGAN *LOW POWER SYSTEM MONITORING* KARBON MONOKSIDA (CO) MENGGUNAKAN METODE *STATE MACHINE***

## **SKRIPSI**

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Teknik

Disusun oleh :  
Raka Bagus Perdana  
NIM : 145150301111044



PROGRAM STUDI TEKNIK KOMPUTER  
JURUSAN TEKNIK INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA  
MALANG  
2018

## PENGESAHAN

PENGEMBANGAN *LOW POWER SYSTEM MONITORING* KARBON MONOKSIDA  
(CO) MENGGUNAKAN *STATE MACHINE*

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Teknik

Disusun Oleh :  
Raka Bagus Perdana  
NIM: 145150301111044

Skripsi ini telah diuji dan dinyatakan lulus pada  
1 Agustus 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Dahnial Syauqy, S.T., M.T., M.Sc.  
NIK: 2016078704231002

Agi Putra Kharisma, S.T., M.T.  
NIK: 2013048604301001

Mengetahui  
Ketua Jurusan Teknik Informatika

Tri Astoto Kurniawan, S.T., M.T., Ph.D.  
NIP: 19710518 200312 1 001

## PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata di dalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 06 Agustus 2018

Raka Bagas Perdana

NIM: 145150301111044



## KATA PENGANTAR

Segala puji dan syukur penulis panjatkan atas rahmat dan karunia-Nya Tuhan yang Maha Esa, sehingga penulis dapat menyelesaikan skripsinya dengan judul ***“PENGEMBANGAN LOW POWER SYSTEM MONITORING KARBON MONOKSIDA (CO) MENGGUNAKAN METODE STATE MACHINE”***.

Penulis menyadari bahwa pembuatan skripsi ini tidak lepas dari bantuan baik motivasi, doa dan jasa dari berbagai pihak. Maka sebab itu, penulis menyampaikan rasa hormat dan terima kasih kepada:

1. Kedua orang tua dan seluruh keluarga yang selalu memberikan motivasi dan semangat serta doa sehingga penulis dapat menyelesaikan skripsinya.
2. Tri Astoto Kurniawan, S.T, M.T, Ph.D. selaku Ketua Jurusan Teknik Informatika
3. Bapak Dahnial Syauqy, S.T., M.T., M.Sc. selaku dosen pembimbing pertama yang telah memberikan dukungan dan bimbingannya kepada penulis untuk segera menyelesaikan skripsi ini.
4. Bapak Agi Putra Karisma, S.T, M.T. selaku dosen pembimbing kedua yang telah memberikan dukungan dalam menyelesaikan laporan skripsi ini.
5. Kekasih yang selalu memberi semangat dalam pengerjaan skripsi Feni Fitriana Dewi.
6. Hanif Yudha Prayoga dan Agung Widya Gumelar sebagai pembimbing yang selalu mendukung dan membantu dalam proses pengerjaan Skripsi.
7. Teman-teman grup *Fake Friends* Sabil, Edo, Pras, Hernanda, Bagus dll. yang telah memberikan dukungan dan segala bentuk bantuan yang telah diberikan.
8. Teman-teman tercinta program studi Teknik Komputer angkatan 2014 yang selalu memberikan semangat. Terima kasih atas semua doa dan dukungan baik dalam bentuk material maupun non material.
9. Dan orang-orang yang selalu mendukung dan mendoakan penulis yang tidak dapat disebutkan satu persatu.

Penulis menyadari bahwa dalam laporan ini masih banyak kekurangan, oleh sebab itu penulis berharap adanya pengembangan lebih lanjut oleh pihak pihak terkait. Semoga laporan ini dapat memberikan manfaat dan referensi untuk melakukan penelitian sebagai langkah penyempurnaan sistem.

Malang, 06 Agustus 2018

Penulis

rakabagasbuduran@gmail.com



## ABSTRAK

Indonesia merupakan negara berkembang yang warganya sangat bergantung dengan adanya kendaraan bermotor. Belakangan ini banyak kasus yang menewaskan secara tiba-tiba di dalam mobil. Penggunaan kendaraan (mobil) dan menurut berita disebabkan oleh keracunan gas Karbon monoksida (CO). Faktor yang menyebabkan terjadinya keracunan bisa dari faktor umur dari kendaraan atau kelalaian pengguna kendaraan itu sendiri. Sering kali pengguna menyalakan kendaraan walaupun sedang terparkir. Dampak yang disebabkan jika kendaraan kurang perawatan bisa membahayakan bagi pengguna kendaraan. Seperti halnya jarang melakukan pengecekan sistem AC mobil dapat menyebabkan asap knalpot dari kendaraan dapat masuk melalui sistem AC dan terhirup oleh pengguna di dalamnya. Dengan adanya alat *monitoring* ini diharapkan dapat mengurangi risiko kematian yang sering terjadi. Untuk dapat membedakan mobil sedang kondisi jalan dan berhenti di gunakan sensor *gyro* yang di letakkan pada *handbrake* mobil. Untuk mengetahui polutan yang ada pada kendaraan menggunakan Sensor MQ-07. Dalam pembuatan sistem monitoring akan dibuat beberapa *state* seperti *state sleep* atau *wake* dan harus dapat berpindah *state* dengan *inputan* dari pengguna sehingga diterapkan *state machine* untuk mekanisme *low power* pada sistem *monitoring* karbon monoksida. Dikarenakan sistem ini akan bekerja dalam jangka waktu yang cukup lama, maka digunakanlah metode *low power* untuk mengurangi beban penggunaan sumber daya dari *power bank*. Kinerja *low power* diperlukan agar sistem dapat bertahan lebih lama karena daya yang digunakan relatif rendah ketika sistem sedang tidak digunakan. Fitur dari *low power* itu sendiri salah satunya adalah *Sleep mode power down* yang mematikan beberapa fungsionalitas ATmega328P. Pengujian arus yang dilakukan menghasilkan, bahwa arus yang digunakan pada sistem berjalan normal adalah 181,02 mA sedangkan ketika sistem dalam keadaan *Sleep* hanya memiliki arus sebesar 138,2 mA. Sehingga sistem berhasil mereduksi arus yang digunakan sebesar 42 mA.

Kata kunci : MQ-7, MPU 6050, *Sleep mode Power down*, *Finite state machine*, ATmega 328p, Mobil

## ABSTRACT

Indonesia is a developing country that citizens rely heavily by the presence of a motor vehicle. Recently many cases which killed suddenly in the car. The use of vehicles (cars) and according to news of the gas poisoning caused by carbon monoxide (CO). The factors that led to the poisoning could be from the age of the vehicle or negligence users vehicles itself. Often the user turning on even though he had parked vehicles. The impact caused if the vehicle less care could harm for users vehicles. Just as rarely do car air conditioning system checking can cause the exhaust fumes from the vehicle can enter through a system of air conditioning and inhaled by users in it. The existence of this monitoring tool is expected to reduce the risk of death is often the case. To be able to differentiate the car are road conditions and stopped at the gyro sensor use in put on the handbrake of automobiles. To find out the pollutants on the vehicle using Sensor MQ-07. In making the monitoring system will be made some state such as sleep or wake state and should be able to switch state with input from users so that applied the state machine for the low power mechanism in carbon monoxide monitoring system. Due to this system will work in quite a long period of time, then the unambiguous method of low power to reduce the burden of resource use power bank. Performance low power is required so that the system can last longer because of the relatively low power used when the system is not in use. Features of low power itself one is Sleep mode power down deadly ATmega328P some functionality. Current testing being performed generates, that the current used on the system running normally is 181.02 mA whereas when the system is in a State of Sleep only has a current of 138.2 mA. So the system is successfully reducing the current use of 42 mA.

**Keywords:** MQ-7, MPU 6050, Sleep mode Power down, Finite state machine, ATmega 328p, Car



## DAFTAR ISI

PENGESAHAN .....	ii
PERNYATAAN ORISINALITAS .....	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
ABSTRACT.....	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR.....	xii
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	2
1.3 Tujuan .....	2
1.4 Manfaat.....	2
1.5 Batasan masalah .....	3
1.6 Sistematika pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN .....	5
2.1 Tinjauan Pustaka .....	5
2.1.1 Implementasi Sistem <i>Monitoring</i> Gas Berbahaya Pada Mobil Berbasis Logika <i>Fuzzy</i> Menggunakan Mikrokontroller. ....	5
2.1.2 Implementasi Low Power Wearable Device Sebagai Heart Rate Monitor Dengan Metode State Machine .....	5
2.2 Dasar Teori.....	6
2.2.1 Pengaruh karbon monoksida Pada tubuh.....	6
2.2.2 Finite State Machine .....	7
2.2.3 Mikrokontroller Atmega 328p .....	8
2.2.4 Sensor MQ 7 .....	9
2.2.5 Sensor <i>Accelero/Gyro</i> MPU 6050.....	10
2.2.6 <i>Buzzer</i> .....	11
2.2.7 LED RGB (Commond catode).....	12
2.2.8 Arduino <i>IDE</i> .....	13
2.2.9 <i>Library</i> .....	14



BAB 3 METODOLOGI .....	15
3.1 Studi Literatur .....	16
3.2 Analisis Kebutuhan .....	16
3.2.1 Kebutuhan Fungsional.....	16
3.2.2 Kebutuhan Non Fungsional.....	16
3.2.3 Analisis Kebutuhan Perangkat Keras.....	17
3.2.4 Analisis Kebutuhan Perangkat Lunak.....	17
3.3 Perancangan Sistem.....	17
3.4 Implementasi Sistem .....	18
3.5 Pengujian Sistem.....	18
3.6 Analisis dan Penarikan Kesimpulan .....	19
BAB 4 REKAYASA DAN KEBUTUHAN SISTEM.....	20
4.1 Gambaran Umum Sistem.....	20
4.1.1 Karakteristik pengguna .....	20
4.1.2 Ruang lingkup .....	20
4.1.3 Lingkungan operasi sistem .....	21
4.1.4 Asumsi dan ketergantungan .....	21
4.2 Kebutuhan Fungsional .....	21
4.3 Kebutuhan Non Fungsional.....	22
4.4 Kebutuhan perangkat keras.....	22
4.5 Kebutuhan perangkat lunak .....	24
BAB 5 PERANCANGAN DAN IMPLEMENTASI .....	25
5.1 Perancangan Sistem.....	25
5.1.1 Perancangan Case Perangkat.....	25
5.1.2 Perancangan Penempatan Posisi Di Kendaraan .....	26
5.1.3 Perancangan Perangkat Keras .....	27
5.1.4 Rangkaian Keseluruhan Sistem .....	32
5.1.5 Perancangan Perangkat Lunak.....	33
5.2 Implementasi Sistem .....	38
5.2.1 Implementasi Perangkat Keras Sistem.....	38
5.2.2 Implementasi Perangkat Lunak.....	42
BAB 6 PENGUJIAN DAN ANALISIS.....	46

6.1. Pengujian Fungsionalitas sensor MPU 6050.....	46
6.1.1 Tujuan.....	46
6.1.2 Prosedur Pengujian .....	46
6.1.3 Hasil dan Analisis.....	46
6.2 Pengujian Fungsionalitas sensor MQ-7 .....	49
6.2.1 Tujuan.....	49
6.2.2 Prosedur Pengujian .....	49
6.2.3 Hasil dan Analisis.....	50
6.3 Pengujian Notifikasi <i>LED RGB</i> .....	52
6.3.1 Tujuan.....	52
6.3.2 Prosedur Pengujian .....	52
6.3.3 Hasil dan Analisis.....	52
6.4 Pengujian Notifikasi <i>Buzzer</i> .....	53
6.4.1 Tujuan.....	54
6.4.2 Prosedur Pengujian .....	54
6.4.3 Hasil dan Analisis.....	54
6.5 Pengujian Arus Ketika Sistem <i>Wake</i> dan <i>Sleep</i> .....	55
6.5.1 Tujuan.....	55
6.5.2 Prosedur Pengujian .....	55
6.5.3 Hasil dan Analisis.....	55
6.6 Pengujian fungsional keseluruhan sistem .....	57
6.6.1 Tujuan.....	57
6.6.2 Prosedur Pengujian .....	58
6.6.3 Hasil dan Analisis.....	58
BAB 7 KESIMPULAN DAN SARAN .....	60
7.1 Kesimpulan.....	60
7.2 Saran .....	60
DAFTAR PUSTAKA.....	62

## DAFTAR TABEL

Tabel 2.1 Persamaan dan Perbedaan Dengan Kajian Pustaka.....	6
Tabel 2.2 Pengaruh karbon monoksida pada Tubuh .....	7
Tabel 2.3 Spesifikasi MPU 6050 .....	11
Tabel 2.4 Spesifikasi pin <i>Buzzer</i> .....	12
Tabel 5.1 Keterangan Pin Perancangan Catu Daya .....	28
Tabel 5.2 Keterangan Pin Perancangan Sensor MPU 6050 .....	29
Tabel 5.3 Keterangan Pin Perancangan Sensor MQ - 7 .....	30
Tabel 5.4 Keterangan pin Perancangan <i>LED RGB</i> .....	31
Tabel 5.5 Keterangan Pin Perancangan Pin <i>Buzzer</i> 5V.....	32
Tabel 5.6 Tabel Penjelasan Elemen <i>State Machine</i> .....	34
Tabel 5.7 Tabel <i>Next State Transition State Machine</i> .....	34
Tabel 5.8 Implementasi Perangkat Lunak <i>Finite state machine</i> .....	42
Tabel 5.9 Potongan Program <i>Motion detection</i> .....	43
Tabel 5.10 Potongan Program MQ-7 .....	44
Tabel 5.11 Potongan Program <i>Power down Sleep</i> .....	45
Tabel 6.1 Tabel Hasil Pengujian Sensor MPU 6050.....	49
Tabel 6.2 Tabel Pengujian Sensor MQ-7 Pada Pembacaan PPM .....	51
Tabel 6.3 Tabel Pengujian Kondisi <i>LED RGB</i> .....	52
Tabel 6.4 Tabel Pengujian <i>Output Buzzer</i> .....	54
Tabel 6.5 Tabel Perbandingan Arus ketika <i>Sleep</i> dan <i>Wake</i> .....	56
Tabel 6.6 Hasil Pengujian Keseluruhan sistem.....	58

## DAFTAR GAMBAR

Gambar 2.1 <i>Finite state machine</i> .....	8
Gambar 2.2 <i>Pin Mapping Mikrokontroller ATmega 328p</i> .....	8
Gambar 2.3 Macam - macam <i>Sleep mode</i> yang ditawarkan Atmega 328p .....	9
Gambar 2.4 Konfigurasi sensor MQ-7 .....	10
Gambar 2.5 <i>Sensor Accelero MPU 6050</i> .....	10
Gambar 2.6 <i>Buzzer</i> .....	11
Gambar 2.7 <i>LED RGB</i> .....	13
Gambar 2.8 <i>Arduino IDE</i> .....	13
Gambar 3.1 Diagram Alir Metodologi .....	15
Gambar 3.2 Diagram Blok Perancangan Sistem .....	17
Gambar 5.1 <i>Case Induk Sistem</i> .....	25
Gambar 5.2 <i>Case Sensor Gyro MPU 6050</i> .....	26
Gambar 5.3 Penempatan posisi dan kondisi tuas <i>Handbrake</i> ditarik.....	26
Gambar 5.4 Penempatan Posisi dan kondisi <i>Handbrake</i> di turunkan .....	27
Gambar 5.5 Perancangan Catu Daya <i>Minsys</i> .....	28
Gambar 5.6 Skematik Sensor MPU 6050 .....	29
Gambar 5.7 Skematis Perancangan Sensor MQ - 7 .....	30
Gambar 5.8 Skematis Perancangan <i>LED RGB</i> .....	31
Gambar 5.9 Skematis Perancangan <i>LED RGB</i> .....	32
Gambar 5.10 Skematis Keseluruhan Alat.....	33
Gambar 5.11 Gambar <i>State Diagram</i> Sistem .....	33
Gambar 5.12 Simulasi <i>Motion Detection</i> pada <i>Handbrake</i> mobil .....	35
Gambar 5.13 <i>Flowchart Motion detection</i> .....	36
Gambar 5.14 <i>Flowchart Sensing</i> Sensor MQ – 7.....	37
Gambar 5.15 <i>Flowchart Power down Sleep</i> .....	38
Gambar 5.16 Implementasi Catu Daya .....	39
Gambar 5.17 Implementasi perangkat keras MPU 6050.....	39
Gambar 5.18 Implementasi Perangkat Keras MQ-7 .....	40
Gambar 5.19 Implementasi <i>LED RGB CC</i> .....	40
Gambar 5.20 Implementasi Perangkat Keras <i>Buzzer</i> .....	41
Gambar 5.21 Implementasi Penempatan Perangkat Keras Pada Mobil .....	41

Gambar 6.1 Pengujian MPU di <i>Handbrake</i> saat ditarik .....	47
Gambar 6.2 Hasil <i>outputan</i> dari Serial <i>Monitor</i> saat <i>Handbrake</i> di Tarik.....	47
Gambar 6.3 Pengujian MPU saat <i>Handbrake</i> didorong.....	48
Gambar 6.4 Hasil <i>outputan</i> dari Serial <i>Monitor</i> saat <i>Handbrake</i> didorong.....	48
Gambar 6.5 Hasil ppm ketika mobil tidak dinyalakan dan belum ada gas .....	50
Gambar 6.6 Hasil ppm ketika mobil sudah nyala dan terdeteksi gas .....	50
Gambar 6.7 Hasil ketika kaca jendela dibuka dan udara terurai .....	51
Gambar 6.8 Pengukuran arus ketika sistem kondisi <i>Wake</i> .....	56
Gambar 6.9 Pengukuran arus ketika sistem kondisi <i>Sleep</i> .....	56



## BAB 1 PENDAHULUAN

### 1.1 Latar belakang

Indonesia merupakan negara berkembang yang warganya sangat bergantung dengan adanya kendaraan bermotor. Belakangan ini banyak kasus yang menewaskan secara tiba-tiba di dalam mobil (Liputan6, 2018). Penggunaan kendaraan (mobil) menurut berita disebabkan oleh keracunan gas Karbon monoksida (CO). Faktor yang menyebabkan terjadinya keracunan bisa dari faktor umur dari kendaraan atau kelalaian pengguna kendaraan itu sendiri. Seringkali pengguna menyalakan kendaraan walaupun sedang terparkir.

Zat karbon monoksida merupakan gas bentuknya tidak terlihat tidak mempunyai warna dan bau yang di hasilkan tidak menyengat sehingga sulit terdeteksi oleh mata awam dan indra lainnya (Wardhana, 2004). Masyarakat tidak pernah menyadari ketika gas sudah terhirup namun efek yang di hasilkan cukup berbahaya.

Dampak yang disebabkan jika kendaraan kurang perawatan bisa membahayakan bagi pengguna kendaraan. Seperti halnya jarang melakukan pengecekan sistem AC mobil dapat menyebabkan asap knalpot dari kendaraan dapat masuk melalui sistem AC dan terhirup oleh pengguna di dalamnya. Ada pula dampak lain yang tidak kalah berbahayanya yaitu kandungan polutan yang dihasilkan dari pembakaran kendaraan tersebut dapat menimbulkan gangguan kesehatan dan dapat juga menimbulkan kematian. Untuk saat ini pencemaran udara yang di hasilkan dari emisi gas buang kendaraan sekitar 70 persen menurut (Munawar, 1999). Kendaraan bermotor di klaim mengeluarkan zat berbahaya sehingga tidak baik untuk kesehatan ataupun di sekitarnya (Kuswara, 2006).

Dengan semakin meningkatnya kebutuhan terdapat beberapa permasalahan. Salah satu permasalahan yang terjadi adalah dengan sumber daya yang terbatas, sistem diharapkan mampu menjalankan berbagai proses. Begitu pula jika sistem digunakan untuk pemantauan secara teratur dan terus-menerus, maka konsumsi energi atau energi yang diperlukan akan bertambah semakin besar. Untuk itulah diperlukan manajemen sumber daya yang baik agar sistem dapat berjalan dalam waktu yang cukup lama dengan sumber daya yang sedikit. Selain dari sisi manajemen sumber daya baterai, penekanan konsumsi energi dapat dilakukan juga dari segi komponen atau *hardware* yang digunakan pada Mikrokontroler. (Pratama, 2017)

Dalam pembuatan sistem monitoring akan dibuat beberapa *state* seperti *sleep* atau *wake* dan harus dapat berpindah *state* dengan *inputan* dari pengguna sehingga diterapkan *state machine* untuk mekanisme *low power* pada sistem *monitoring* karbon monoksida. *State Machine* merupakan pemodelan dari



perilaku sebuah sistem atau obyek yang kompleks dengan beberapa kondisi atau mode yang terdefiniskan dimana mode transisi berubah sesuai dengan keadaan (Matahari Bhakti Nendya, 2015). Maka digunakan *state machine* sebagai mekanisme *power* pada sistem *monitoring* karbon monoksida. Dengan adanya mekanisme *low power* yang diterapkan dapat dilakukan *monitoring* karbon monoksida secara berkala tanpa memerlukan banyak waktu untuk mengisi ulang baterai dari perangkat karena konsumsi daya yang rendah.

Berdasarkan studi kasus yang di ambil dari pembahasan di atas maka perlu adanya sistem yang dapat *memonitoring* untuk mengatasi keberadaan polutan yang masuk ke dalam kabin kendaraan roda empat tersebut dan sistem menggunakan metode *Finite State Machine* untuk perpindahan *state* saat jalannya program. Selama ini penelitian hanya sebatas *monitoring* sehingga sistem perlu perkembangan lebih lanjut mengenai sumber dayanya, sehingga diharapkan sistem dapat lebih sempurna. Untuk penelitian kali ini akan di sisipkan *Low Power System* agar sistem lebih hemat daya dan tidak mengganggu kelistrikan di dalam mobil.

## 1.2 Rumusan masalah

Berdasarkan uraian latar belakang di atas, didapatkan rumusan masalah yang akan dijelaskan sebagai berikut :

1. Bagaimana melakukan perancangan dan pengimplementasian mekanisme *low power mode* kepada sistem *monitoring* CO berdasarkan sensor MPU 6050 dan sensor MQ – 7?
2. Bagaimana melakukan perancangan dan pengimplementasian metode *Finite State Machine* pada sistem *monitoring* CO?
3. Bagaimana melakukan pengukuran performa mekanisme *low power mode* dan *Finite State Machine* yang di terapkan kepada sistem *monitoring* CO ?

## 1.3 Tujuan

Berdasarkan dari latar belakang yang Sesuai di atas, maka tujuan dari penelitian ini sebagai berikut :

1. Dapat merancang dan mengimplementasikan mekanisme *low power mode* pada sistem berdasarkan sensor MPU 6050 dan sensor MQ – 7.
2. Dapat merancang dan mengimplementasikan metode *Finite State Machine* pada sistem berdasarkan sensor MPU 6050 dan sensor MQ – 7.
3. Dapat mengukur performa mekanisme *low power mode* terhadap sistem *monitoring* CO berdasarkan sensor MPU 6050 dan Sensor MQ - 7

## 1.4 Manfaat

Manfaat yang didapat dari studi kasus ini adalah sebagai berikut :

1. Penerapan *low power mode* diharapkan dapat meringankan beban tegangan agar lebih rendah dan lebih efisiensi sumber daya.



2. Mengimplementasikan mekanisme *low power* dapat membuat alat menjadi lebih lama penggunaannya dan tidak memerlukan pengisian sumber daya terus menerus.
3. Menghindari terjadinya kematian secara tiba-tiba yang disebabkan oleh polutan Karbon monoksida.

### 1.5 Batasan masalah

Batasan digunakan agar penelitian tidak terlalu meluas sehingga peneliti dapat lebih fokus pada suatu masalah. Untuk Batasan masalah bisa dilihat pada poin di bawah ini.

1. Penelitian ini menggunakan parameter udara karbon monoksida (CO)
2. Data hanya bisa ditampilkan di *LED RGB*
3. Untuk mendapatkan data polutan dari MQ – 7, sistem harus dalam kondisi *wake*
4. Peneliti hanya membuat alat sebatas *prototype*
5. Sistem ini menggunakan *Finite state machine* dalam pemrosesan data
6. Sensor MPU 6050 hanya dapat digunakan ketika mobil kondisi di jalan datar.

### 1.6 Sistematika pembahasan

Sistematika penulisan dalam skripsi ini adalah sebagai berikut :

#### BAB 1 Pendahuluan

Bab ini berisi tentang latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah, sistematika pembahasan dan jadwal pelaksanaan dari penelitian dan sistematika penulisan laporan berjudul “ Pengembangan Low Power System Monitoring CO Menggunakan Metode State Machine”.

#### BAB 2 Landasan Kepustakaan

Pada bab ini akan dijelaskan mengenai teori-teori dan ilmu pendukung sebagai landasan pengerjaan penelitian.

#### BAB 3 Metodologi

Bab ini menjelaskan tentang langkah-langkah kerja yang dilakukan pada penelitian ini, seperti studi literatur, analisis kebutuhan, perancangan sistem dan implementasi sistem.

#### BAB 4 Rekayasa Kebutuhan

Bab ini menjelaskan tentang deskripsi umum sistem, kebutuhan perangkat keras dan lunak serta kebutuhan fungsional sistem.

#### BAB 5 Perancangan dan Implementasi

Bab ini menjelaskan tentang perancangan dan implementasi berupa perangkat keras dan perangkat lunak yang digunakan pada penelitian ini.

**BAB 6 Pengujian dan Analisis**

Bab ini menjelaskan tentang langkah kerja dalam melaksanakan pengujian sistem dan analisis pengujian yang dilakukan.

**BAB 7 Penutup**

Bab ini menjelaskan tentang kesimpulan dan saran yang didapat dari hasil hasil pengujian dan analisis.



## BAB 2 LANDASAN KEPUSTAKAAN

Pada bab ini berisikan tentang tinjauan pustaka yang menjabarkan daftar pustaka dan dasar teori yang dibutuhkan untuk keperluan penelitian. Kajian pustaka itu sendiri membahas tentang penelitian yang sebelumnya. Sedangkan dasar teori di gunakan untuk membahas teori yang di perlukan untuk keperluan penelitian.

### 2.1 Tinjauan Pustaka

Di sini akan di lakukan pembahasan mengenai penelitian sebelumnya yang dapat di gunakan sebagai landasan kepastakaan yang berkaitan dengan penelitian yang akan datang.

#### 2.1.1 Implementasi Sistem *Monitoring* Gas Berbahaya Pada Mobil Berbasis Logika *Fuzzy* Menggunakan Mikrokontroller.

Fendy Purwanto pernah melakukan penelitian untuk membahas tentang Perancangan dan Implementasi alat *monitoring* gas berbahaya yang berbasis logika *fuzzy* untuk menentukan nilai kualitas udara di dalam mobil tempat alat *monitoring* itu diletakkan. Peneliti menggunakan 2 sensor yaitu sensor MQ-7 sebagai *inputan* yang berfungsi mengambil data polutan karbon monoksida (CO), dan sensor MQ 135 sebagai *inputan* yang berfungsi mengambil data polutan gas Amonia (NH<sub>3</sub>) (Purwanto, 2016).

Pada penelitian tersebut, lebih fokus terhadap hasil data yang di peroleh dari metode *Fuzzy Sugeno*. Perbedaan dari penelitian ini dengan skripsi ini yaitu pada konsumsi daya mikrokontroller yang didesain agar lebih minimum.

#### 2.1.2 Implementasi Low Power Wearable Device Sebagai Heart Rate Monitor Dengan Metode State Machine .

Ihsanurrahim pernah melakukan penelitian tentang Implementasi *Low Power Wearable Device* Sebagai *Heart Rate* Dengan Metode *State Machine*. Dalam penelitian ini peneliti menerapkan *Low Power System* untuk *wearable devicenya* dikarenakan perlunya penghematan konsumsi daya agar perangkat dapat bekerja dalam jangka waktu yang panjang dengan tenaga baterai. Mikrokontroller yang di gunakan peneliti adalah tipe Atmega 328 karena dapat digunakan sebagai *low power*.

Pada konsumsi daya purwarupa sistem dapat mereduksi konsumsi daya sebesar 23,78 mA atau sebesar 75,06 % dari 31,68 mA dengan menerapkan *state machine* dan *Power down Sleep* sebagai mekanisme *low power*. Uji kinerja purwarupa memiliki jarak efektif hingga 12 meter dan waktu yang dibutuhkan untuk *Wake* dari *Sleep* adalah 0,0508 detik (Ihsannurrahim, 2018). Berikut Tabel perbandingan kajian pustaka peneliti sebelumnya dapat dilihat pada Tabel 2.1

**Tabel 2.1 Persamaan dan Perbedaan Dengan Kajian Pustaka**

No.	Penulis	Judul	Persamaan	Perbedaan
1.	Fendy Purwanto	Implementasi Sistem <i>Monitoring</i> Gas Berbahaya Pada Mobil Berbasis Logika <i>Fuzzy</i> Menggunakan Mikrokontroller.	Sensor yang digunakan pada penelitian ini sama-sama menggunakan sensor gas MQ-7.	Penelitian ini terfokus pada hasil olahan data yang di olah menggunakan <i>fuzzy sugeno</i> .
2.	Ihsannurrahim	Implementasi <i>Low Power Wearable Device</i> Sebagai Heart Rate Monitor Dengan Metode <i>State Machine</i> .	Sama menggunakan metode <i>low power</i> dan <i>state machine</i> untuk perpindahan step nya.	Peneliti menggunakan sensor <i>Pulse</i> untuk pengukur jantung. Dan komunikasi data menggunakan <i>Bluetooth</i>

## 2.2 Dasar Teori

Dalam sub bab ini akan dijelaskan referensi dasar teori sebagai pengetahuan tentang desain interaksi dari disiplin ilmu interaksi manusia dan komputer serta macam-macam kebutuhan komponen dalam pembuatan sistem ini.

### 2.2.1 Pengaruh karbon monoksida Pada tubuh

Karbon monoksida merupakan gas yang tidak terlihat oleh kasat mata dan tidak berbau. Menurut (Soedomo, 2003) Karbon monoksida dapat mempengaruhi kesehatan dan menimbulkan gejala penyakit seperti sesak nafas, penurunan daya tampung darah untuk oksigen. Kemungkinan terburuk jika terlalu banyak yang di hisap masuk ke dalam paru- paru maka akan menghalangi Oksigen masuk ke dalam paru-paru. Karbon monoksida itu sendiri merupakan gas senyawa racun metabolis sehingga dapat langsung menghambat peredaran darah. (Yulianti, et al, 2013). Menurut penelitian (Wardhana, 2004) bahwa kandungan CO yang terhirup oleh manusia mempunyai Batasan PPM nya yang dapat dilihat pada tabel 2.2

**Tabel 2.2 Pengaruh karbon monoksida pada Tubuh**

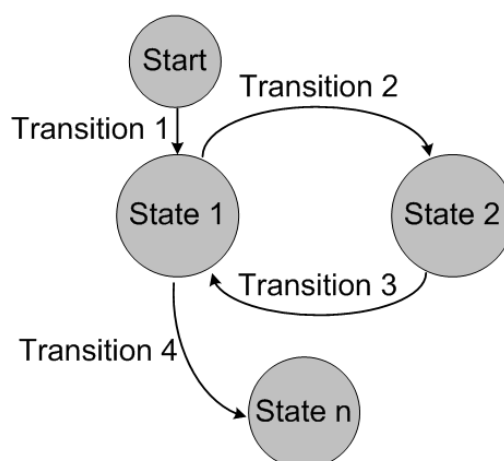
Konsentrasi CO di udara	Gangguan Pada Tubuh
3 Ppm	Tidak ada
5 Ppm	Belum Begitu Terasa
10 Ppm	System Saraf Sentral
20 Ppm	Panca Indera
40 Ppm	Fungsi Jantung
60 Ppm	Sakit Kepala
80 Ppm	Sulit Bernafas
100 Ppm	Pingsan – Kematian

Sumber : (Wardhana, 2004)

### 2.2.2 Finite State Machine

FSM didefinisikan sebagai perangkat komputasi yang memiliki *input* berupa *string* dan *output* yang merupakan satu dari dua nilai yang dapat di-*accept* dan *reject* (Rich, 2009). Sebuah *finite-state machine* (fsm) atau *finite automata*, *State machine* adalah sebuah mesin abstrak yang dapat berada pada tepat satu kondisi dari beberapa jumlah *state* yang ditentukan. Fsm dapat berpindah dari satu *state* ke *state* lainnya berdasarkan respon dari *inputan* dan perubahan dari *state* ke *state* lainnya disebut dengan *transition*. sebuah fsm didefinisikan sebagai sebuah daftar dari beberapa *state*, *initial state*, dan kondisi dari tiap transisi. Fsm dapat juga dikatakan sebagai pemodelan dari perilaku (*behavior*) sebuah sistem atau obyek yang kompleks dengan beberapa kondisi atau mode yang terdefiniskan dimana mode transisi berubah sesuai dengan keadaan. FSM terdiri dari empat elemen utama:

1. *State* yang mendefinisikan kelakuan dan mungkin menghasilkan aksi.
2. Transisi *state* di mana merupakan perpindahan dari satu *state* ke *state* lain.
3. Aturan atau kondisi yang harus dipenuhi supaya ada *transisi state* kejadian(*events*).
4. *Input* yang terjadi baik *internal* maupun *external*, yang memungkinkan *trigger* aturan dan mengacu ke transisi *state*.



**Gambar 2.1 Finite state machine**

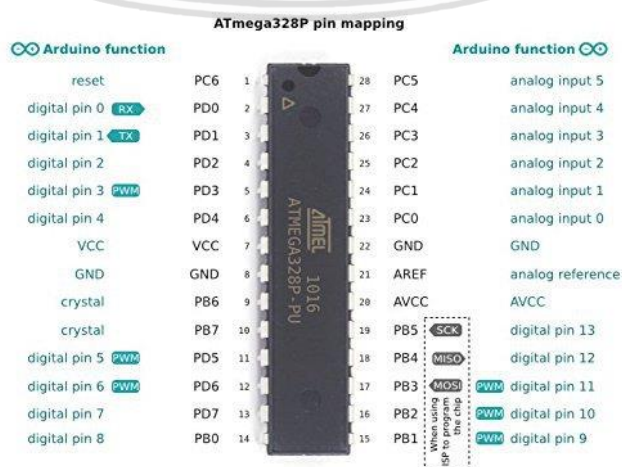
Sumber : (Oracle, n.d.)

### 2.2.3 Mikrokontroler Atmega 328p

Atmel AVR core merupakan kombinasi dari banyak set instruksi dengan 32 register. Semua 32 register terhubung langsung dengan *Arithmetic Logic Unit (ALU)*, memungkinkan dua register dapat diakses dalam sebuah instruksi yang dieksekusi dalam satu *clock cycle*. Arsitektur yang dihasilkan menjadi lebih efisien dalam menjalankan *code* selagi memperoleh nilai *outputan* hingga sepuluh lebih cepat dibandingkan dengan mikrokontroler CISC konvensional.

ATmega328p memiliki fitur *low power* yaitu salah satunya adalah *Power-down Sleep*. *Power-down* mode menyimpan konten dari register namun menghentikan *Oscillator*, mematikan fungsi lain dari *chip* hingga *interrupt* selanjutnya atau reset *hardware*. Untuk melihat *pin mapping* atmega dapat dilihat pada gambar 2.2.

**Gambar 2.2 Pin Mapping Mikrokontroler ATmega 328p**



Sumber : (Microchip Technology Inc,2016)



ATmega 328p mempunyai beberapa fitur berupa mekanisme Low Power yang dapat digunakan untuk pengembangan Perangkat cerdas yang membutuhkan efisiensi sumber daya dikarenakan untuk pengembangan yang lebih lanjut seharusnya fitur perangkat cerdas lebih banyak namun sumber daya yang digunakan tidak terus menerus menguras kapasitas. Fitur *low power* yang akan digunakan merupakan tipe *power down* karena menggunakan *external interrupt* untuk pemacu sistem *wake* nya. Masih banyak fitur yang ditawarkan *low power* untuk menjawab keperluan pengembang. Pilihan *low power* dapat dilihat pada Gambar 2.3 dibawah ini.

Sleep Mode	Active Clock Domains					Oscillators		Wake-Up Sources							
	clk <sub>cpu</sub>	clk <sub>flash</sub>	clk <sub>io</sub>	clk <sub>adc</sub>	clk <sub>asy</sub>	Main Clock Source Enabled	Timer Oscillator Enabled	INT and PCINT	TW Address Match	Timer2	SPM/EEPROM Ready	ADC	WDT	Other I/O	Software BOD Disable
Idle			Yes	Yes	Yes	Yes	Yes <sup>(2)</sup>	Yes	Yes	Yes	Yes	Yes	Yes	Yes	
ADC Noise Reduction				Yes	Yes	Yes	Yes <sup>(2)</sup>	Yes <sup>(3)</sup>	Yes	Yes <sup>(2)</sup>	Yes	Yes	Yes		
Power-Down								Yes <sup>(3)</sup>	Yes				Yes		Yes
Power-Save				Yes			Yes <sup>(2)</sup>	Yes <sup>(3)</sup>	Yes	Yes			Yes		Yes
Standby <sup>(1)</sup>						Yes		Yes <sup>(3)</sup>	Yes				Yes		Yes
Extended Standby				Yes <sup>(2)</sup>		Yes	Yes <sup>(2)</sup>	Yes <sup>(3)</sup>	Yes	Yes			Yes		Yes

Gambar 2.3 Macam - macam *Sleep mode* yang ditawarkan Atmega 328p

Sumber: (Microchip Technology Inc., 2016)

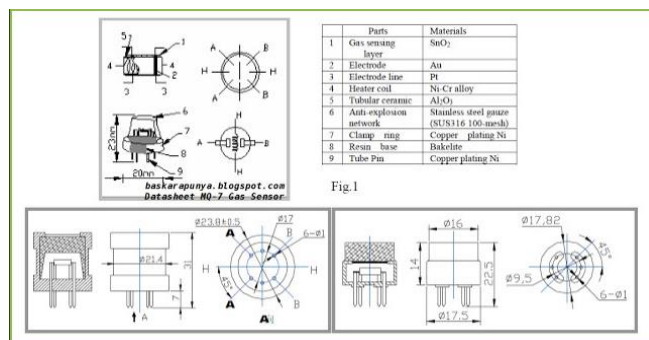
#### 2.2.4 Sensor MQ 7

MQ 7 merupakan sensor gas yang digunakan dalam peralatan untuk mendeteksi gas karbon monoksida (CO) dalam kehidupan sehari-hari, industri, atau mobil. Fitur dari sensor gas MQ 7 ini adalah mempunyai sensitivitas yang tinggi terhadap karbon monoksida (CO), stabil, dan berumur panjang. Sensor ini menggunakan catu daya *heater* : 5V AC/DC dan menggunakan catu daya rangkaian : 5 VDC, jarak pengukuran : 20 – 2000 ppm untuk ampu mengukur gas karbon monoksida. Untuk konfigurasi sensor dapat dilihat pada Gambar 2.4 dibawah.

Kondisi Standar Sensor Bekerja

- VCC yang digunakan =  $5V \pm 0.1$
- VH *Heater* saat Tinggi  $5V \pm 0.1$
- VH (L)/ *Heater* saat kondisi *low* =  $1.4V \pm 0.1$
- Resistansi dapat diatur sesuai kebutuhan
- RH Resistansi Pemanas=  $33\Omega \pm 5\%$
- TH (H) Waktu Pemanasan (Tinggi) =  $61 \pm 1$  seconds
- TH (L) Waktu Pemanasan (Rendah) =  $91 \pm 1$  seconds





**Gambar 2.4 Konfigurasi sensor MQ-7**

Sumber : (Datasheet MQ 7)

### 2.2.5 Sensor Accelero/Gyro MPU 6050

MPU 6050 adalah sebuah sensor IMU (*Inertial Measurement Unit*) yang dapat mendeteksi pergerakan dari benda berdasarkan sumbu X, Y, dan Z. MPU 6050 terdiri dari 6-DOF yaitu 3-axis sensor akselerometer untuk mendeteksi percepatan dan 3-axis sensor *gyroscope* untuk mendeteksi sudut. MPU 6050 bergerak konstan maju dan mundur yang berada dalam susunan material *piezo electric crystal*. Ketika dimiringkan, *crystal* dari *piezo electric* menerima gaya searah dengan kecenderungan arah miringnya DAN menyebabkan inersia bergerak. *Crystal* kemudian menghasilkan arus yang kemudian diperkuat. Hasil dari pergerakan kemudian diolah oleh mikrokontroller. Pada Gambar 2.5 adalah bentuk fisik dari MPU-6050. Spesifikasi dapat dilihat pada Tabel 2.3.



**Gambar 2.5 Sensor Accelero mpu 6050**

(Sumber : (Arduino, 2014)

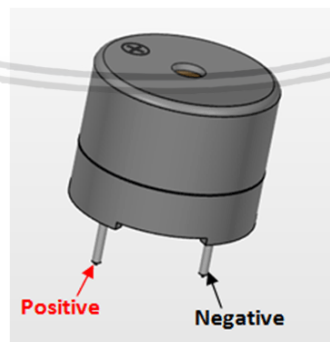
**Tabel 2.3 Spesifikasi MPU 6050**

Part	MPU 6050
VDD	2.375V-3.46V
VLOGIC	1.71V to VDD
Pin 8	VLOGIC
Serial Interface	I2C
Pin 9	AD0
Pin 23	SCL
Pin 24	SDA

### 2.2.6 Buzzer

*Buzzer* adalah sebuah komponen elektronika yang berfungsi untuk mengubah getaran listrik menjadi getaran suara. Pada dasarnya prinsip kerja *buzzer* hampir sama dengan *loud speaker*, jadi *buzzer* juga terdiri dari kumparan yang terpasang pada diafragma dan kemudian kumparan tersebut dialiri arus sehingga menjadi elektromagnet, kumparan tadi akan tertarik ke dalam atau keluar, tergantung dari arah arus dan polaritas magnetnya, karena kumparan dipasang pada diafragma maka setiap gerakan kumparan akan menggerakkan diafragma secara bolak-balik sehingga membuat udara bergetar yang akan menghasilkan suara. *Buzzer* biasa digunakan sebagai indikator bahwa proses telah selesai atau terjadi suatu kesalahan pada sebuah alat.

*Buzzer* adalah perangkat elektronika yang terbuat dari elemen *piezoceramics* pada suatu diafragma yang mengubah getaran/vibrasi menjadi gelombang suara. *Buzzer* menggunakan resonansi untuk memperkuat intensitas suara. Untuk melihat bentuk fisik dari *Buzzer* dapat dilihat pada gambar 2.6 dibawah ini. Untuk spesifikasi dapat dilihat pada Tabel 2.4.



**Gambar 2.6 Buzzer**

(Sumber : (Components 101, t.thn.))

Tabel 2.4 Spesifikasi pin *Buzzer*

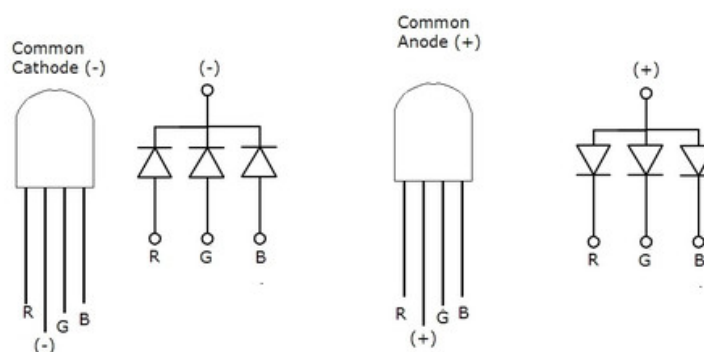
Simbol	Deskripsi
<i>Positive</i>	<i>Voltage 3,3 V – 5V</i>
<i>Negative</i>	<i>Ground</i>

### 2.2.7 LED RGB (Common cathode)

Lampu LED suatu jenis komponen elektronika yang memancarkan cahaya ketika sinyal *HIGH* dan akan mati jika diberi *inputan LOW* Light Emitting Diode atau sering disingkat dengan LED adalah komponen elektronika yang dapat memancarkan cahaya monokromatik ketika diberikan tegangan maju. LED merupakan keluarga Dioda yang terbuat dari bahan semikonduktor. Warna Cahaya yang akan di pancarkan oleh LED tergantung pada jenis bahan semikonduktor yang dipergunakannya. Bentuk LED mirip dengan sebuah bohlam (bola lampu) yang kecil dan dapat dipasangkan dengan mudah ke dalam berbagai perangkat elektronika.

Berbeda dengan lampu pijar, LED tidak memerlukan pembakaran filamen sehingga tidak menimbulkan panas dalam menghasilkan cahaya. LED merupakan keluarga dari dioda yang terbuat dari semikonduktor. Cara kerjanya pun hampir sama dengan dioda yang memiliki dua kutub yaitu kutub positif (P) dan kutub negatif (N). LED hanya akan memancarkan cahaya apabila dialiri tegangan maju (bias *forward*) dari anoda menuju ke katoda.

LED terdiri dari sebuah *chip* semikonduktor yang di doping sehingga menciptakan *junction* P dan N. Yang dimaksud dengan proses doping dalam semikonduktor adalah proses untuk menambahkan ketidakmurnian (*impurity*) pada semikonduktor yang murni sehingga menghasilkan karakteristik kelistrikan yang diinginkan. Ketika LED dialiri tegangan maju atau bias *forward* yaitu dari anoda (P) menuju ke katoda (K), kelebihan elektron pada N-Type material akan berpindah ke wilayah yang kelebihan *hole* (lubang) yaitu wilayah yang bermuatan positif (P-Type material). Saat elektron berjumpa dengan *hole* akan melepaskan *photon* dan memancarkan cahaya monokromatik (satu warna).

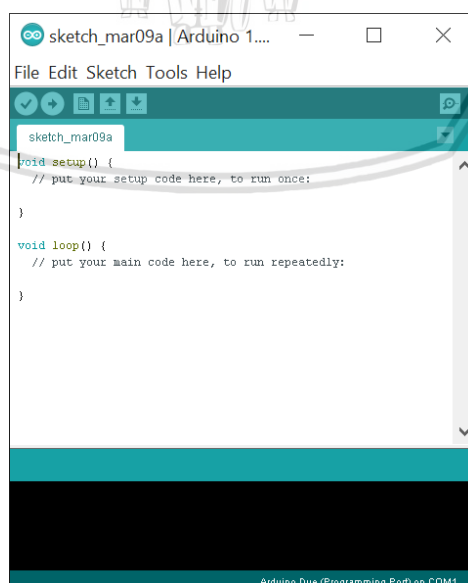


**Gambar 2.7 LED RGB**

Sumber : (DEVICE, t.thn.)

### 2.2.8 Arduino IDE

Arduino IDE adalah aplikasi untuk melakukan pemrograman pada Arduino dan untuk mengeksekusi program tersebut ke Arduino. IDE dari Arduino tidak memiliki fitur – fitur canggih seperti *debugger* atau *code completion*. Arduino IDE terdiri dari editor, *compiler*, *loader*, dan monitor serial. Namun terdapat fitur *auto formatting* agar baris program menjadi rapi. Informasi yang akan di dapat berupa pesan area yang memberikan umpan balik saat menyimpan program maupun menjalankan program serta menampilkan informasi apabila terdapat kode program yang masih mengalami kesalahan dalam pembacaannya. Untuk mengatasi permasalahan dapat menggunakan menu *Help* yang di sediakan. IDE ini dapat berjalan di berbagai macam *operating system* dan bisa di dapatkan secara *online* di situs resmi Arduino secara *free*. Adapun bentuk IDE dapat dilihat pada gambar 2.8.



**Gambar 2.8 Arduino IDE**

Sumber: (Arduino, 2014)

### 2.2.9 Library

*Library* adalah kumpulan program atau fungsi yang telah ada pada *compiler* atau *intepreter* untuk memudahkan pemrogram membuat program dan tidak perlu mengakses langsung sistem komputer untuk memprogram. *Library* yang disediakan bergantung pada bahasa pemrograman yang digunakan maupun compiler atau intepreter yang digunakan, untuk mengetahui pustaka apa saja yang ada perlu dilakukan eksplorasi terhadap bahasa pemrograman yang digunakan. Setiap pustaka menyediakan fungsi-fungsi tertentu yang dapat mendukung pemrograman, oleh karena itu lakukan eksplorasi terlebih dahulu, jika fungsi telah ada pada pustaka kita tidak perlu lagi membuatnya sendiri, cukup menggunakan fungsi-fungsi yang ada pada pustaka. *Library* yang akan di gunakan pada penelitian ini adalah sebagai berikut.

#### A. *Avr Sleep ( Sleep.h)*

*Library* yang digunakan untuk menerapkan fungsi yang akan digunakan untuk membuat program *Sleep* dari mikrokontroller.

#### B. *Avr Power (Power.h)*

*Library* ini digunakan untuk memuat set fungsi yang akan digunakan membuat program mengaktifkan dan menonaktifkan fitur dari mikrokontroller seperti ADC, timer, dan lainnya.

#### C. *Library MPU 6050*

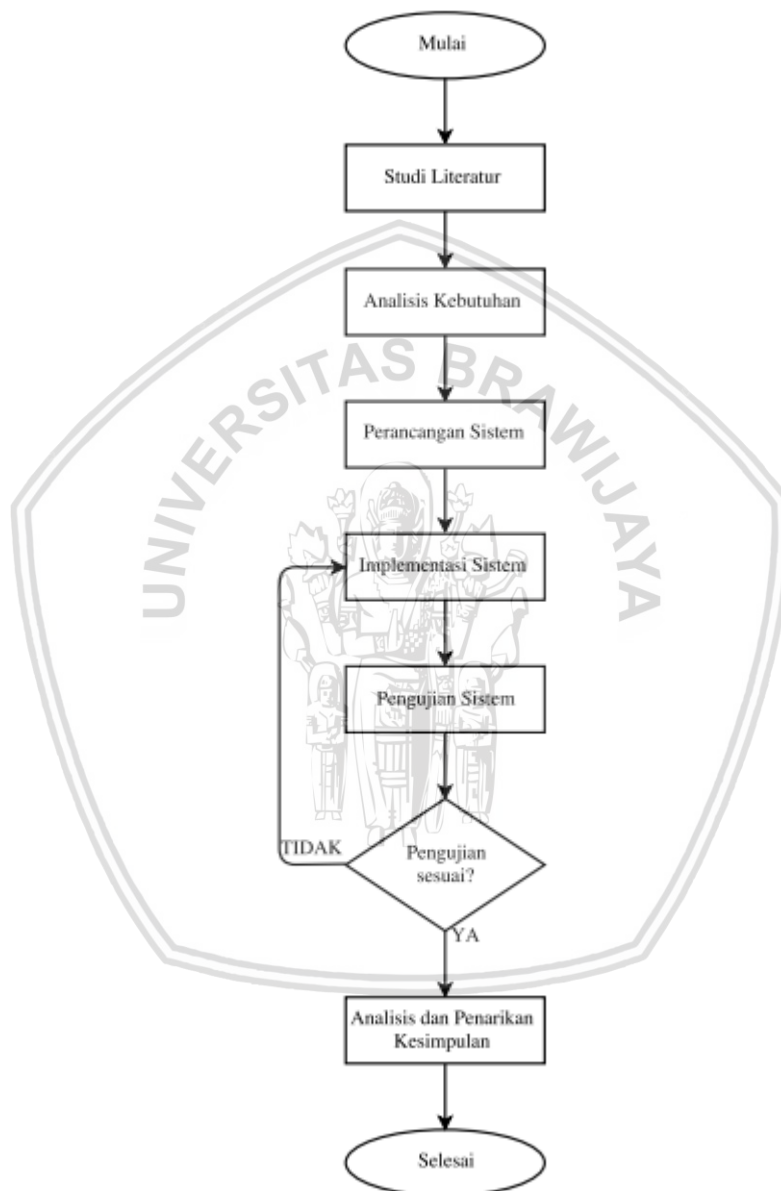
*Library* ini digunakan untuk memuat set fungsi yang digunakan untuk membuat program mengirim data pada memori register sensor MPU 6050. Pada set fungsi terdapat beberapa fungsi yang dapat digunakan untuk membuat program mengatur pembacaan sensor, mengatur *interrupt*, *internal filter*, dan lain-lain.

#### D. *Library I2c (Wire.h)*

*Library* ini digunakan untuk berkomunikasi dengan perangkat I2c/TWI yang di gunakan untuk pin SDA dan SCL yang digunakan pada sensor mpu 6050.

### BAB 3 METODOLOGI

Pada bab ini menjelaskan tentang langkah-langkah penelitian serta metodologi dalam penulisan skripsi. Langkah-langkah yang terdapat dalam penelitian ini dapat dilihat pada gambar 3.1.



Gambar 3.1 Diagram Alir Metodologi



### 3.1 Studi Literatur

Pada penelitian ini, dilakukan studi literatur yang dikumpulkan merupakan dasar - dasar teori dan juga penelitian sebelumnya yang digunakan sebagai bahan acuan dan dasar dari penelitian yang akan dilakukan. Studi literatur mengenai Karbon monoksida yang akan di deteksi pada sistem *monitoring* yang akan dirancang. Pencarian dan studi dari hasil penelitian sebelumnya membantu untuk mendapatkan gambaran jelas mengenai bagian - bagian yang akan dikerjakan. Dasar teori - teori yang dijadikan pendukung untuk penelitian yang akan dilakukan akan dijadikan sebagai penguat untuk melakukan perancangan dalam penelitian dimana dasar teori – teori yang akan digunakan adalah karbon monoksida, *Finite state machine*, ATmega328P, Sensor MQ-7, Sensor MPU 6050, *Buzzer* , dan *LED RGB*, *Arduino IDE*, *Library*.

### 3.2 Analisis Kebutuhan

Analisis kebutuhan dibutuhkan untuk mengidentifikasi dan analisis apa yang dibutuhkan untuk perancangan sebuah sistem. Kebutuhan analisis dibedakan menjadi dua bagian yaitu kebutuhan fungsional dan kebutuhan non fungsional. Adapun kebutuhan akan dijelaskan sebagai berikut:

#### 3.2.1 Kebutuhan Fungsional

Kebutuhan fungsional adalah suatu kebutuhan tentang apa saja yang dapat dilakukan sistem, untuk mengetahui apa saja fungsi yang harus dicapai dari penelitian ini. Berikut kebutuhan fungsional yang terdapat pada penelitian ini:

1. Sistem dapat menentukan kondisi kendaraan sedang berhenti atau berjalan.
2. Sistem dapat mengetahui berapa banyak polutan yang ada di dalam kabin mobil.
3. Sistem dapat memberi informasi kadar CO melalui LED RGB.
4. Sistem dapat memberi peringatan sesuai kadar CO yang di deteksi sistem melalui *Buzzer*.

#### 3.2.2 Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional merupakan bagian yang menjadi batasan yang harus dilakukan ketika melakukan penelitian untuk sistem yang akan dibangun. kebutuhan non fungsionalitas dalam penelitian yang akan dilakukan adalah :

1. Arduino IDE digunakan untuk membuat program yang akan di *upload* ke mikrokontroler ATmega328P serta sebagai *compiler* yang dapat di *upload* ke mikrokontroler.
2. Pemrograman dilakukan dengan bahasa c++ dalam instruksi Arduino.
3. Program yang dibuat diunggah ke mikrokontroler menggunakan Arduino IDE.



### 3.2.3 Analisis Kebutuhan Perangkat Keras

Analisis kebutuhan perangkat keras perlu di lakukan agar mengetahui perangkat keras apa saja yang digunakan beserta fungsi yang digunakan. Kebutuhan perangkat keras yang dibutuhkan adalah sebagai berikut:

1. Mikrokontroller ATmega 328p sebagai modul pemrosesan yang memiliki fitur *Low Power* sehingga dapat mereduksi arus.
2. Sensor *Gyro* MPU sebagai *input* kondisi kendaraan sedang berhenti atau berjalan.
3. Sensor MQ 7 digunakan sebagai *inputan* saat kondisi sistem *Wake*.
4. untuk memberi peringatan dini saat polutan mulai melebihi ambang batasnya.
5. *LED RGB* sebagai sarana informasi intensitas polutan yang tertangkap sistem.

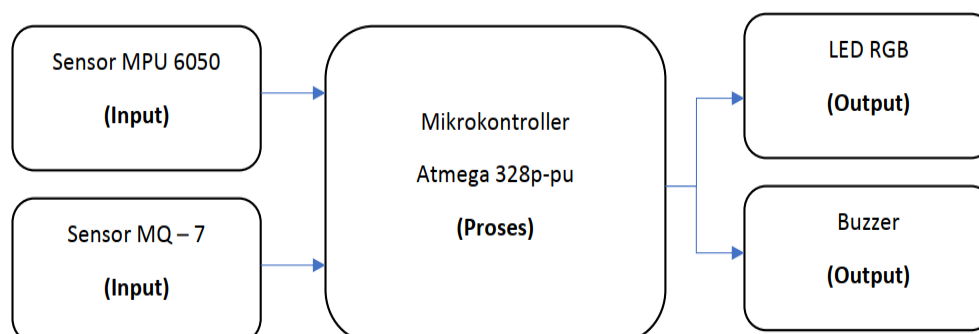
### 3.2.4 Analisis Kebutuhan Perangkat Lunak

Analisis kebutuhan perangkat lunak perlu dilakukan untuk mengetahui perangkat lunak apa saja yang digunakan beserta fungsi yang digunakan. Kebutuhan perangkat lunak yang dibutuhkan adalah sebagai berikut:

1. Dibutuhkan Arduino *IDE* untuk wadah membuat program yang akan di *upload* ke mikrokontroller ATmega328P dan sebagai compiler yang dapat di *upload* ke mikrokontroller.
2. Program yang digunakan menggunakan Bahasa C++ dalam aturan Arduino. Program yang telah dibuat diunggah ke mikrokontroller menggunakan Arduino *IDE*.

## 3.3 Perancangan Sistem

Pada subbab ini merupakan bagian dimana perancangan dari sistem yang akan dibuat dibahas. Setelah semua kebutuhan dari sistem telah didapat maka akan dilakukan perancangan sistem sesuai dengan diagram blok yang dapat dilihat pada Gambar 3.2.



Gambar 3.2 Diagram Blok Perancangan Sistem

Berdasarkan gambar 3.2 ada tiga tahapan pada perancangan sistem *monitoring* yang dapat akan di jelaskan pada poin berikut :

1. Sensor MPU 6050 dan Sensor MQ-7 sebagai *Inputan*.
2. *Mikrokontroller* sebagai otak sistem, seluruh proses di olah mikrokontroller ATmega 328p.
3. *LED RGB* dan *Buzzer* merupakan *output* sarana informasi dan peringatan yang dapat dilihat oleh penumpang mobil untuk mengetahui kondisi polutan saat sistem di jalankan.

### 3.4 Implementasi Sistem

Implementasi berisikan beberapa langkah yang diambil agar sistem dapat terwujud sesuai dengan desain sistem atau perancangan sistem. Berdasarkan perancangan sistem, implementasi dapat meliputi hal-hal berikut :

1. Instalasi perangkat lunak seperti Arduino IDE .
2. Merancang sistem berdasarkan *software* nya dengan Arduino IDE, ketika sistem akan sleep dan wake.
3. Merancang sistem berdasarkan hardware-nya dimana hardware ini terdiri dari 1 Sensor MPU 6050, 1 Sensor gas MQ 7, 1 LED RGB, 1 Mikrokontroler ATmega328P, 2 Kapsitor 22 pf dan 1 Crystal Osilator. Sensor MPU 6050 dan sensor gas menjadi *hardware inputan* untuk ATmega 328P, lalu LED akan menjadi *outputan*, Kapasitor dan Kristal Osilator akan menjadi perangkat yang akan membantu ATmega328P bekerja. Serta dirancang bentuk *prototype* dari sistem.

### 3.5 Pengujian Sistem

Pengujian dan analisis sistem dilakukan untuk menguji tingkat keberhasilan sistem dalam menentukan peringatan kepada pengguna berdasarkan *inputan* yang ada, pengujian-pengujiannya adalah :

1. Pengujian pertama dilakukan untuk menguji fungsionalitas sensor MPU 6050 saat di gunakan pada *Handbrake* mobil.
2. Pengujian kedua adalah menguji fungsionalitas sensor MQ- 7 saat menangkap polutan di dalam kabin mobil.
3. Pengujian ketiga adalah notifikasi LED RGB ketika polutan terdeteksi sistem dan LED menyala sesuai parameter yang di tentukan.
4. Pengujian keempat adalah notifikasi *Buzzer* ketika polutan terdeteksi oleh sistem melebihi batas aman yang telah di tentukan sistem.
5. Pengujian kelima adalah pengukuran arus sistem ketika kondisi *Wake* dan kondisi *Sleep*.

6. Pengujian keenam adalah pengujian keseluruhan fungsionalitas sistem ketika sensor MPU 6050 menjadi *input* dan semua elemen sistem dapat bekerja sesuai yang di harapkan.

### 3.6 Analisis dan Penarikan Kesimpulan

Analisis dilakukan untuk menguji tingkat keberhasilan sistem akan dilakukan setelah pengujian pada sistem. Kesimpulan merupakan jawaban dari rumusan masalah yang telah dibahas sebelumnya. Dari hasil analisis dan penarikan kesimpulan nantinya akan memunculkan saran untuk penelitian selanjutnya.



## BAB 4 REKAYASA DAN KEBUTUHAN SISTEM

Pada bab 4 akan dijelaskan secara lebih detail mengenai rekayasa kebutuhan yang harus dipenuhi dalam proses perancangan hingga proses implementasi sistem. Kebutuhan yang dimaksud meliputi gambaran umum dari sistem, kebutuhan fungsional dan kebutuhan non fungsional.

### 4.1 Gambaran Umum Sistem

Sistem yang akan dibangun adalah suatu mekanisme untuk memaksimalkan penggunaan catu daya sebuah alat *monitoring* (CO) karbon monoksida dalam kabin mobil. Metode yang akan digunakan adalah *State Machine* dan menggunakan atau memanfaatkan fitur yang terdapat di dalam mikrokontroller, sehingga alat bisa dibedakan menjadi beberapa *state* yaitu dalam keadaan *Sleep*, *standby* dan lain - lain, sehingga daya baterai yang digunakan dapat se efisien mungkin atau direduksi arusnya. Untuk stimulusnya agar dapat berpindah dari satu *state* ke *state* yang lain menggunakan sensor *gyro* yang terdapat di sensor MPU 6050.

Untuk *Output* dalam sistem berikut dapat diketahui melalui *LED RGB* yang menampilkan informasi seberapa bahayakah kandungan (CO) karbon monoksida yang terdapat di dalam kabin mobil berupa cahaya yang berbeda sesuai dengan tingkatan kondisinya. LED berwarna hijau menandakan kondisi kabin mobil aman dari gas polutan, LED menyala berwarna biru menandakan polutan di dalam kabin mobil tercemar dengan tingkat kategori menengah, sedangkan jika warna LED berubah menjadi warna merah, maka sistem menandakan polutan yang ada di dalam kabin sangat berbahaya. Dengan kondisi yang sangat berbahaya, sistem bertugas memberi peringatan melalui *buzzer*, agar penumpang menyadari adanya gas berbahaya dan segera melakukan tindakan penyelamatan segera mungkin.

#### 4.1.1 Karakteristik pengguna

Karakteristik pengguna diperuntukkan bagi yang mempunyai kendaraan mobil yang ingin mengetahui kondisi kesehatan komponen (AC) *air conditioning* mobil sehingga terhindar dari bahaya polutan CO yang masuk ke dalam kabin mobil.

#### 4.1.2 Ruang lingkup

Ruang lingkup sistem atau lingkungan sistem yang berjudul “Pengembangan *Low Power System Monitoring* Karbon Monoksida menggunakan metode *State Machine*” adalah :

1. Sistem bergantung pada *inputan* Sensor MPU 6050 sebagai awal mula bekerjanya sistem.

2. Sistem diletakkan pada bagian *console* tengah samping jok depan agar mendekati *handbrake* dan hembusan AC (*Air Conditioning*) untuk penumpang belakang dapat terdeteksi oleh sistem.

#### 4.1.3 Lingkungan operasi sistem

Lingkungan operasi sistem merupakan ketergantungan sistem ketika di berikan tegangan agar berjalan dengan semestinya. Sistem dapat bekerja dengan *range* 1,8 V sampai 5V. Sumber tegangan tidak bisa menyala jika tegangan kurang dari 1,8 V dan tidak lebih dari 5,5 V agar tidak terbakar.

#### 4.1.4 Asumsi dan ketergantungan

Beberapa asumsi dan ketergantungan persyaratan sistem sebagai berikut:

1. Data sensor dapat dibaca sesuai yang di inginkan pembuat apa bila peletakkannya tepat dan tidak berubah ubah.
2. Untuk dapat menyala diperlukan sumber daya yang cukup stabil antara 1,8 V – 5,5 V jika dibawah 1,8 V maka mikrokontroller bisa mati dan jika diatas 5,5 V maka komponen yang ada bisa rusak karena kelebihan muatan.

#### 4.2 Kebutuhan Fungsional

Pada subbab ini akan dijelaskan mengenai kebutuhan fungsional dari sistem, kebutuhan fungsional dari sistem yang akan dibuat merupakan penjelasan dari fitur atau fungsi yang harus ada dan dapat diperoleh dengan sistem tersebut. Pada kebutuhan sistem meliputi lingkup *inputan* dan *outputan* sistem, serta fungsi respons sistem terhadap *inputan* dan *outputan* dalam proses bekerjanya sistem. Kebutuhan fungsional dari sistem adalah sebagai berikut:

##### 1. Sensor MPU 6050 dapat membaca kondisi kendaraan sedang berjalan atau berhenti.

Fungsi ini berguna agar sistem dapat menerapkan mekanisme *low power* karena sensor tersebut bertindak sebagai *interrupt* pemicu sistem supaya berpindah *state* dari *Sleep* ke *Wake* atau sebaliknya.

##### 2. Sensor MQ 7 dapat membaca kadar polutan CO Karbon monoksida.

Fungsi sensor ini sebagai *input* ke 2 yang berguna sebagai pendeteksi adanya polutan yang terperangkap di dalam kabin mobil dapat terdeteksi dan dapat diolah sistem.

##### 3. LED RGB dapat memberi informasi kondisi kadar polutan di dalam kabin.

Fungsi LED ini merupakan sumber informasi sistem agar penumpang dapat mengetahui kondisi polutan yang ada di dalam kabin. Warna cahaya berwarna hijau menandakan kondisi polutan sedang kondisi aman, warna cahaya berwarna biru menandakan polutan sedang kondisi sedang, jika warna merah maka bisa

dipastikan polutan sudah memenuhi kabin mobil dan statusnya sudah berbahaya.

#### 4. **Buzzer dapat memberi peringatan kepada pengguna kendaraan.**

*Buzzer* berfungsi sebagai *output* sistem terakhir yang bertugas sebagai peringatan kepada penumpang. *Buzzer* berbunyi “*beep*” terus menerus sampai kandungan polutan dinyatakan aman atau kondisi sedang.

### 4.3 Kebutuhan Non Fungsional

Kebutuhan non fungsional merupakan kebutuhan yang digunakan untuk mendukung fungsi dasar dari sistem yang akan dibuat agar dapat berjalan dengan optimal. kebutuhan non fungsionalitas dalam penelitian yang akan dilakukan, adalah:

1. Arduino IDE digunakan untuk membuat program yang akan diunggah ke mikrokontroler ATmega328P serta sebagai *compiler* yang dapat diunggah ke mikrokontroler.
2. Pemrograman dilakukan dengan bahasa c++ dalam instruksi Arduino.
3. Program yang dibuat diunggah ke mikrokontroler menggunakan Arduino IDE

### 4.4 Kebutuhan perangkat keras

Kebutuhan perangkat keras merupakan kebutuhan yang harus dipenuhi yang berwujud perangkat keras. Pada penelitian ini dibutuhkan beberapa perangkat keras, kebutuhan perangkat keras yang digunakan adalah sebagai berikut:

#### 1. **Mikrokontroler ATmega 328p**

Mikrokontroler *ATmega 328p* merupakan otak dari *sistem*, kebutuhan yang di harus dimiliki oleh mikrokontroler yaitu mempunyai mekanisme *low power*. Agar sesuai dengan yang di butuhkan oleh sistem.

#### 2. **Kristal 16 Mhz**

Kristal merupakan komponen pendukung yang digunakan untuk menjalankan sebuah minimum system. Kristal di sebut juga dengan *oscillator* (Pembangkit Frekuensi) komponen ini berfungsi seperti layaknya jantung jika di analogikan pada tubuh manusia. Padan mikrokontroler yang digunakan saat ini hanya bisa menggunakan 16 Mhz sesuai dengan *requirement* dari Atmega 328p.

#### 3. **Kapasitor 22pf**

Kapasitor 22pf merupakan *requirement* yang juga di butuhkan ketika membangun minimum *system* adalah komponen yang berfungsi menyimpan dan melepaskan muatan listrik. Dalam rangkaian ini kapasitor juga sebagai penyaring (filter) yang digunakan pada *power supply*. Komponen ini digunakan untuk



menghasilkan tegangan output yang lebih halus sehingga tegangan DC dapat lebih konstan.

#### **4. Sensor MQ 7**

Sensor MQ 7 merupakan perangkat keras yang fungsi kerjanya di dalam sistem sebagai *input*, karakteristik sensor ini peka terhadap kandungan polutan CO karbon monoksida.

#### **5. Sensor MPU 6050**

Sensor ini merupakan sensor *Accelero*. Fungsi Sensor ini untuk mendeteksi kendaraan sedang kondisi berjalan atau berhenti dengan cara menentukan 3 *axis accelero*. Sensor ini bekerja *nonstop* saat sistem mulai dinyalakan.

#### **6. LED RGB (CC) Common Catode**

LED RGB merupakan sumber informasi satu satunya yang digunakan oleh sistem ini agar mekanisme *low power* benar benar maksimal, karena hanya membutuhkan catu daya rendah namun cukup berguna dalam sarana informasi sebuah sistem.

#### **7. Buzzer 5 V**

Buzzer merupakan sebuah komponen yang biasanya digunakan untuk peringatan atau *output* sebuah sistem, dalam penelitian ini buzzer digunakan sebagai *output* suara di saat sistem mendeteksi adanya polutan berbahaya di dalam kabin mobil.

#### **8. PCB Matrix**

PCB berguna untuk menjadi wadah berkumpulnya dari beberapa komponen yang digunakan oleh sistem. PCB yang digunakan dalam penelitian ini adalah PCB Matrix yang biasa digunakan untuk latihan dalam membuat berbagai macam elektronika.

#### **9. Power Bank 5 V**

Sebagai sumber daya tegangan yang masuk pada sistem agar dapat berjalan. Power bank yang digunakan menggunakan tegangan 5 V.

#### **10. Laptop**

Sebagai pemberi kontrol pemrograman pada ATmega328p melalui *software IDE* dan *monitoring* peneliti pada serial monitor.



## 4.5 Kebutuhan perangkat lunak

Kebutuhan perangkat lunak adalah kebutuhan untuk mengetahui perangkat lunak apa saja yang dibutuhkan dalam membangun sistem. Kebutuhan perangkat lunak meliputi :

### 1. *Integrated development environment (IDE)*

Perangkat lunak *integrated development environment (IDE)* adalah sebuah perangkat lunak yang digunakan sebagai pengembangan atau memprogram di dalam lingkup Arduino. Arduino itu sendiri menggunakan bahasa C++ sehingga banyak yg menggunakan *software* tersebut yang di klaim mudah digunakan.

### 2. *Library avr Sleep (Sleep.h)*

*Library* yang digunakan untuk menerapkan fungsi yang akan digunakan untuk membuat program *Sleep* dari mikrokontroler.

### 3. *Library avr power (Power.h)*

*Library* ini digunakan untuk memuat set fungsi yang akan digunakan membuat program mengaktifkan dan menonaktifkan fitur dari mikrokontroler seperti *ADC*, *timer*, dan lainnya.

### 4. *Library Comunication I2c (wire.h)*

*Library* ini digunakan untuk berkomunikasi dengan perangkat I2c/TWI yang di gunakan untuk pin SDA dan SCL yang digunakan pada sensor MPU 6050.

### 5. *Library MPU 6050*

*Library* ini digunakan untuk memuat set fungsi yang digunakan untuk membuat program mengirim data pada memori register sensor MPU 6050. Pada set fungsi terdapat beberapa fungsi yang dapat digunakan untuk membuat program mengatur pembacaan sensor, mengatur *interrupt*, *internal filter*, dan lain lain.

### 6. *Bootloader*

Sebuah program kecil yang ditanamkan di sebuah mikrokontroller ATmega 328p sebagai penanda dalam menentukan proses *input* dan *output* yang akan digunakan saat program atau *sketch* di jalankan melalui *software IDE*.

## BAB 5 PERANCANGAN DAN IMPLEMENTASI

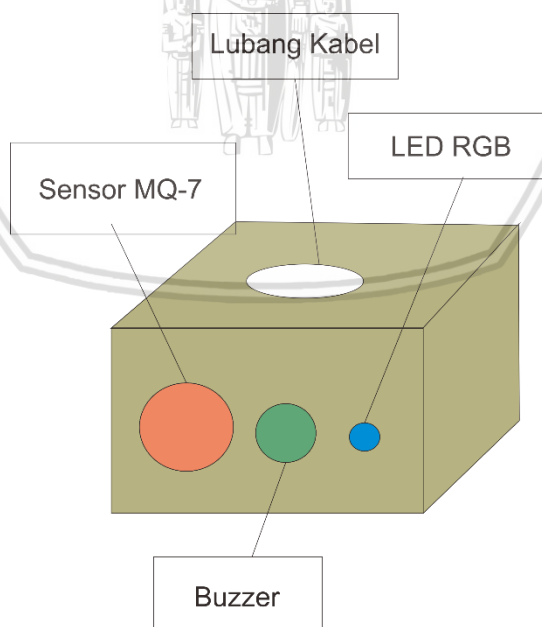
Pada bab perancangan dan implementasi akan dijelaskan lebih detail mengenai bagaimana sistem dirancang atau dibuat baik secara *hardware* maupun *software* serta bagaimana sistem diimplementasikan metodenya. Sistem juga akan dirancang ke dalam bentuk *prototype* alat sehingga sistem dapat berjalan sesuai dengan prinsip kerja sistem.

### 5.1 Perancangan Sistem

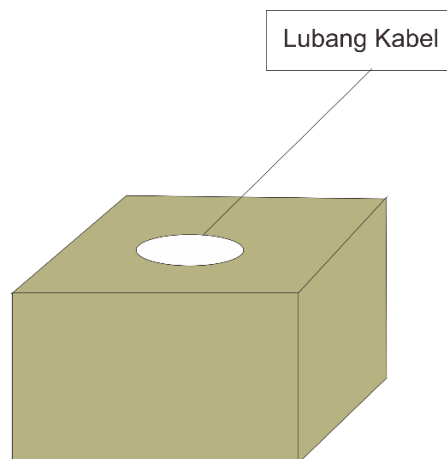
Pada bab ini akan menjelaskan mengenai perancangan dan implementasi dari sistem *low power* menggunakan metode *state machine* dan *Power down Sleep* dengan memanfaatkan pergerakan *handbrake* untuk kembali ke posisi *Wake*. Perancangan yang dimaksud berupa perancangan perangkat keras dan juga perangkat lunak dari sistem dan implementasi dari sistem yang telah dirancang.

#### 5.1.1 Perancangan Case Perangkat

Perancangan *case* perangkat merupakan tempat yang akan digunakan untuk berkumpulnya rangkaian sistem agar lebih terlihat rapi. *Case* di bagi 2 tempat yaitu *case* induk sistem yang terdiri dari sensor MQ – 7, LED RGB dan *Minimum System* dan *case* yang kedua merupakan *case* untuk sensor Gyro MPU 6050 yang letaknya harus terpisah dari induk sistemnya. Bentuk dari *case* itu sendiri yaitu berupa kubus yang panjang dan lebarnya sudah diukur sebelumnya. Contoh perancangan *case* dapat dilihat pada gambar 5.1 dan 5.2 .



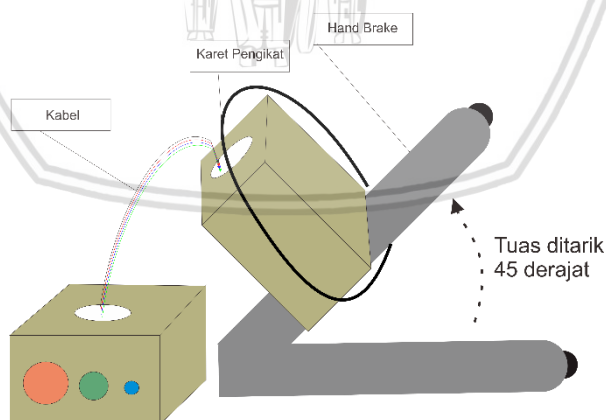
Gambar 5.1 Case Induk Sistem



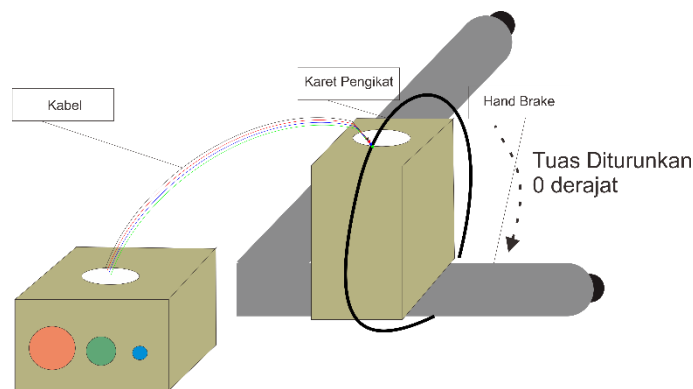
Gambar 5.2 Case Sensor Gyro MPU 6050

### 5.1.2 Perancangan Penempatan Posisi Di Kendaraan

Perancangan Penempatan posisi di kendaraan merupakan bagian dari perancangan sistem karena sistem dapat diletakkan pada satu tempat tertentu agar sesuai dengan yang diinginkan oleh pengguna. Untuk sensor MPU 6050 di letakkan pada *Handbrake* mobil agar dapat mendeteksi kondisi mobil sedang berjalan atau tidak. Sedangkan *case* induk sistem di letakkan di belakang *handbrake* agar kabel penghubung sistem dengan sensor MPU 6050 tidak terlalu jauh dan diharapkan dapat menangkap semburan AC di dalam mobil. Contoh penempatan posisi handbrake ketika di tarik dapat dilihat pada gambar 5.3 dan untuk penempatan posisi *handbrake* ketika di dorong dapat dilihat pada gambar 5.4.



Gambar 5.3 Penempatan posisi dan kondisi tuas *Handbrake* ditarik



Gambar 5.4 Penempatan Posisi dan kondisi *Handbrake* di turunkan

### 5.1.3 Perancangan Perangkat Keras

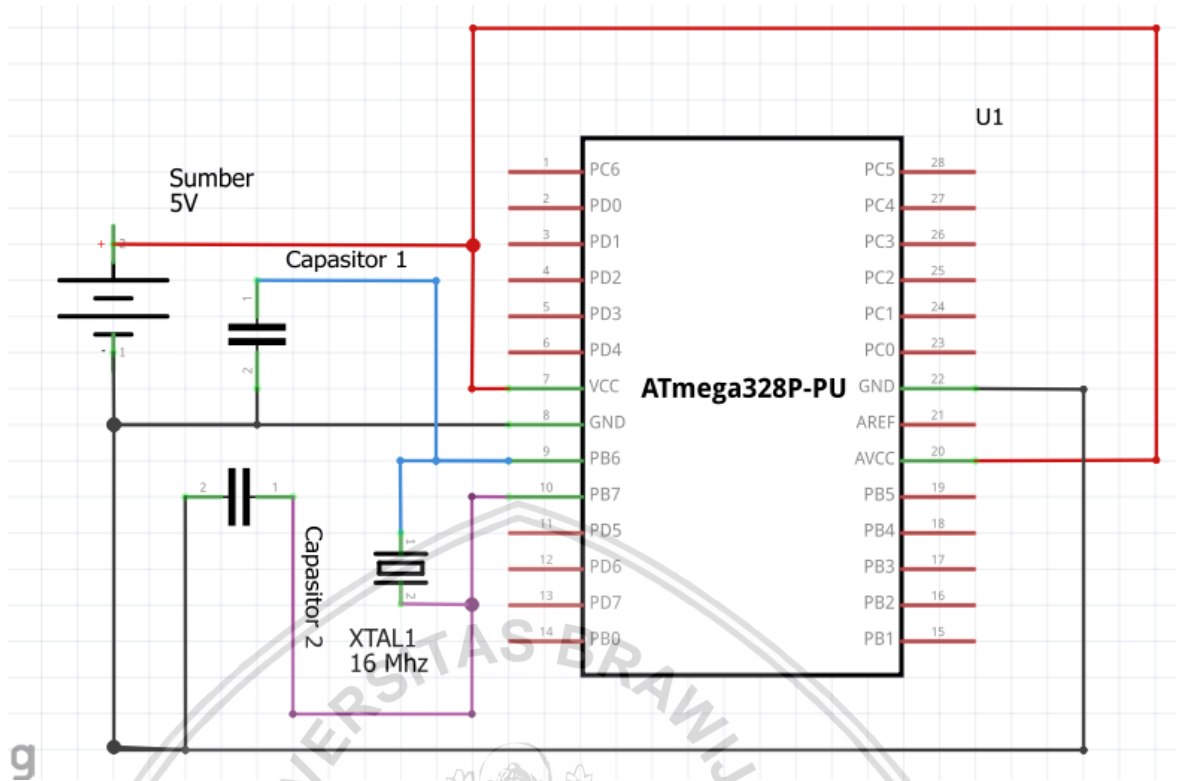
Pada subbab ini merupakan bagian-bagian dari perancangan perangkat keras dari sistem yang akan digunakan.

#### 5.1.3.1 Perancangan Subsistem Perangkat Keras

Pada bagian ini akan di jelaskan macam macam bagian dari sebuah sistem yang kemudian akan di jadikan menjadi satu agar menjadi sebuah sistem yang menyatu.

##### a. Perancangan Perangkat Keras Catu Daya dan Minsys

Pada sistem menggunakan baterai sebagai tenaga pembangkit, sumber yang digunakan merupakan *power bank* dengan tegangan 5V dan kapasitas 5000mAh. Untuk menghubungkan baterai ke *portable device* menggunakan *connector* dengan kutub positif untuk pin VCC dan negatif untuk pin GND. untuk merancang Atmega328p agar bisa digunakan sebagai mana fungsinya dengan beberapa komponen pendukung yaitu *crystal* 16 Mhz yang berfungsi sebagai pemompa data yang bersifat timer, jika di analogikan yaitu layaknya fungsi jantung pada tubuh manusia. 2 Kapasitor 22uF yang berfungsi menyimpan dan melepas muatan listrik. Push Button berfungsi untuk tombol *reset*, ketika tombol ini di tekan maka mikrokontroller akan membaca program mulai dari awal. Skema untuk pemasangan catu daya dengan menggunakan *power bank* dapat dilihat pada gambar 5.5 dan Tabel 5.1 merupakan pin yang digunakan.



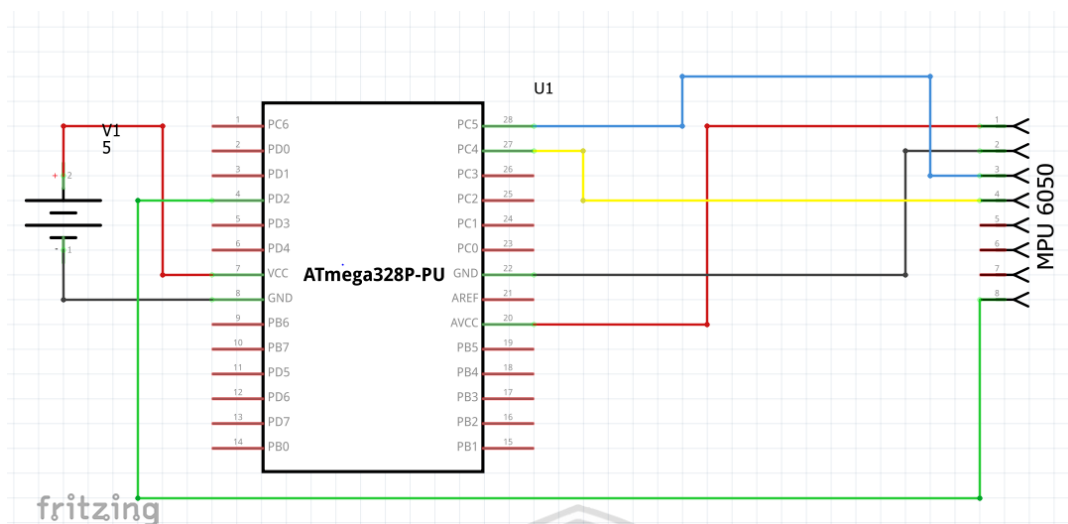
Gambar 5.5 Perancangan Catu Daya *Minsys*

Tabel 5.1 Keterangan Pin Perancangan Catu Daya

Power Bank	Atmega 328p	Keterangan
Positif (Merah)	Pin 7	VCC
Negatif (Hitam)	Pin 8	GND
Positif (Merah)	Pin 20	AVCC
Negatif (Hitam)	Pin 22	AGND

#### b. Perancangan Perangkat Keras MPU 6050

MPU 6050 dapat memberikan acuan *Wake* pada sistem digunakan sensor untuk membaca gerakan yaitu sensor MPU 6050. Sistem memerlukan *Interface* TWI / I2C untuk terhubung ke MPU 6050, pin yang digunakan adalah pin A4/SDA data, pin A5/SCL untuk *Clock*, lalu sebagai *interrupt* untuk *Wake* dihubungkan dengan pin D2 sebagai INT0, pin VCC ke pin VCC dan pin GND ke pin GND untuk memberikan daya sehingga MPU 6050 dapat bekerja. *Interrupt* yang digunakan pada MPU 6050 merupakan int mode LOW. *Interrupt* itu sendiri merupakan suatu program yang fungsinya untuk menghentikan sejenak program yang telah berjalan sebelumnya dan melakukan perintah *interrupt* itu sendiri. Untuk skema perancangan *Motion detection* dapat dilihat pada Gambar 5.6 dan tabel 5.2 merupakan keterangan pin yang digunakan.



Gambar 5.6 Skematik Sensor MPU 6050

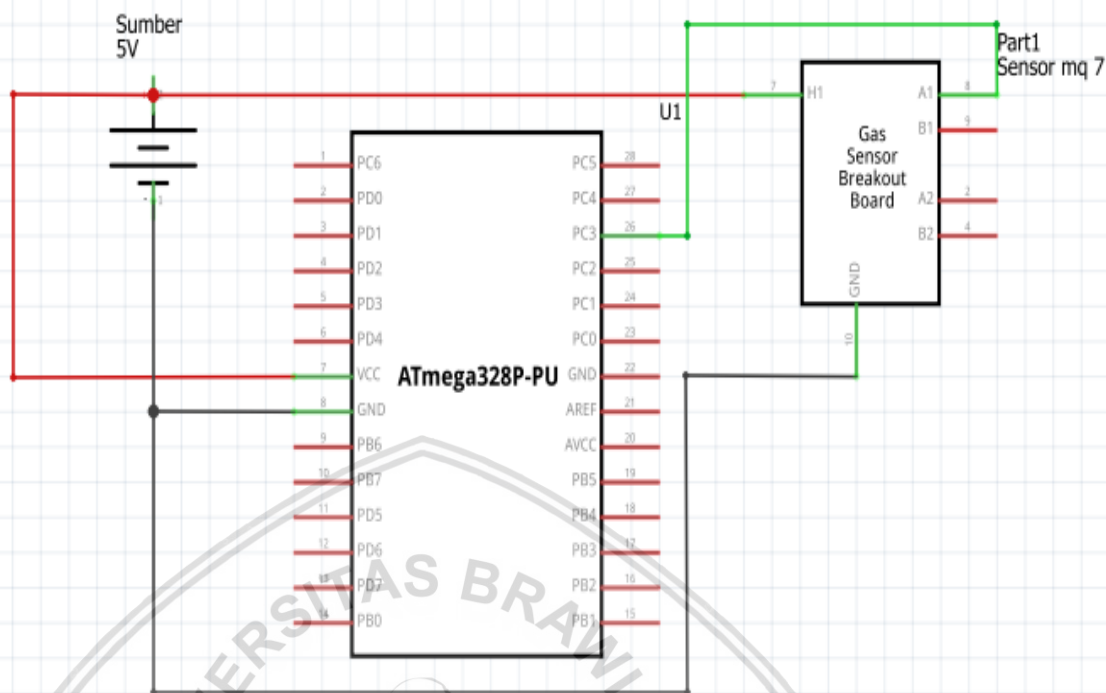
Tabel 5.2 Keterangan Pin Perancangan Sensor MPU 6050

MPU 6050	ATmega328p	Keterangan
VCC	Pin 7	VCC
GND	Pin 8	GND
SCL	Pin 28	SCL/ANALOG 5
SDA	Pin 27	SDA/ANALOG 4
INT	Pin 4	INTERUPT 0/ DIGITAL PIN 2

### c. Perancangan Perangkat Keras sensor MQ-7

Sistem *monitoring* yang dirancang bertujuan untuk membaca polutan yang masuk ke dalam kabin mobil, untuk polutan gas karbon monoksida digunakan sensor MQ-7. Sensor MQ-7 merupakan sensor *analog* yang dapat bekerja pada tegangan 5V. Sensor agar dapat bekerja harus di hubungkan ke pin VCC ke pin VCC, pin GND ke pin GND dan pin *output* pada pin A3. Skema untuk perancangan perangkat keras sensor MQ-7 dapat dilihat pada Gambar 5.7 dan tabel 5.3 merupakan pin yang di gunakan dalam perancangan.





**Gambar 5.7 Skematis Perancangan Sensor MQ - 7**

**Tabel 5.3 Keterangan Pin Perancangan Sensor MQ - 7**

Sensor MQ-7	ATmega 328p	Keterangan
VCC	Pin 7	VCC
GND	Pin 8	GND
Data	Pin 23	ANALOG 3

Karena sensor gas yang di gunakan bertipe *semi conductor* maka di dalam mikrokontroler harus menggunakan pin ADC. ADC itu sendiri merupakan converter dari sinyal analog yang di olah menjadi sinyal digital. Berikut contoh perhitungan nilai ADC secara manual

$$ADC = \frac{V_{in}}{V_{ref}} \times 1023$$

$V_{in}$  = Output nilai tegangan yang di dapatkan dari sensor.

$V_{ref}$  = Merupakan tegangan referensi yaitu bernilai 5V.

1023 = Data maksimal output dari Atmega 328p yang bernilai 10 bit.

Untuk permisalan  $V_{in}$  yang di gunakan dalam contoh bernilai 4,9 V maka perhitungan yang didapat adalah

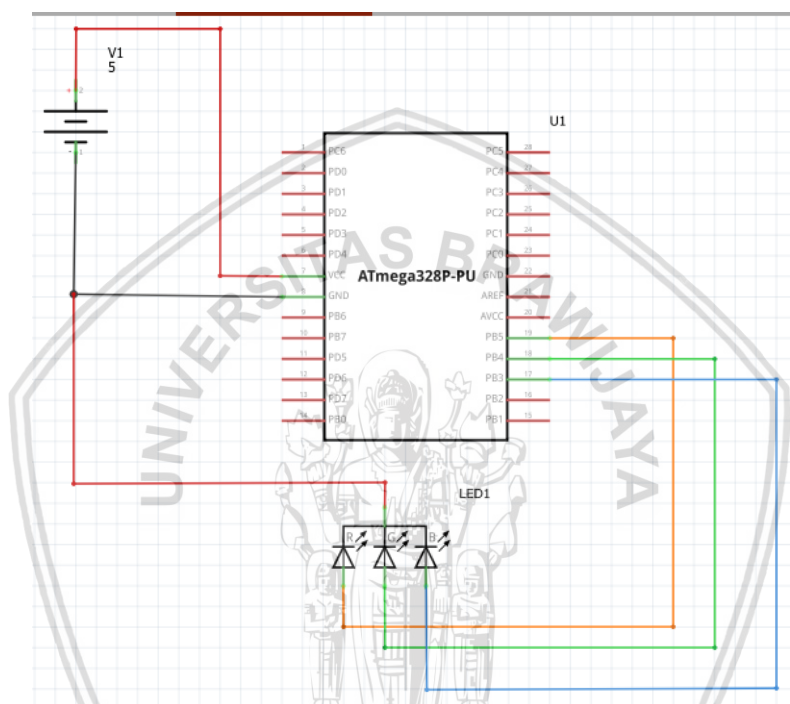
$$ADC = \frac{4,9}{5} \times 1023$$

$$ADC = 1.002,54$$

Maka nilai ADC maksimal yang bisa di hasilkan dari pin ADC senilai 1.003 karena di bulatkan sehingga hasil biner yang di dapatkan 111101011.

#### d. Perancangan Perangkat keras **LED RGB CC (Common Catode)**

Untuk memudahkan pengguna mengamati hasil pembacaan sensor dan melihat kondisi polutan digunakan **LED RGB** untuk menampilkan hasil dari pembacaan sensor MQ-7. **LED RGB** bekerja saat kondisi sistem dalam keadaan *state Wake* , pin yang diperlukan untuk dihubungkan yaitu pin *red* ke pin *digital* 13, pin *green* ke pin *digital* 12, pin *blue* ke pin *digital* 11 dan pin **GND** ke pin **GND**. Skema untuk perancangan perangkat keras tampilan dapat dilihat pada Gambar 5.8 dan Tabel 5.4 yang digunakan pada perancangan **LED RGB**.



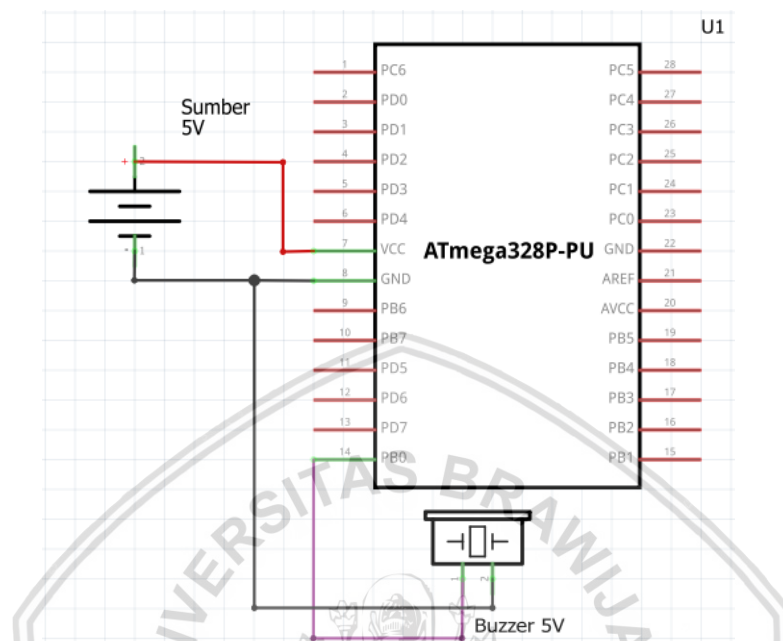
Gambar 5.8 Skematis Perancangan **LED RGB**

Tabel 5.4 Keterangan pin Perancangan **LED RGB**

<b>LED RGB</b>	<b>ATmega 328p</b>	<b>Keterangan</b>
<i>Red</i>	Pin 19	Pin <i>digital</i> 13
<i>Green</i>	Pin 18	Pin <i>digital</i> 12
<i>Blue</i>	Pin 17	Pin <i>digital</i> 11
<b>GND</b>	Pin 8	<b>GND</b>

#### e. Perancangan Perangkat keras **BUZZER 5V**

Untuk memudahkan pengguna mengetahui hasil pembacaan sensor MQ-7 dan mendengar peringatan sistem mengenai kondisi polutan di dalam mobil menggunakan **Buzzer**. **Buzzer** bekerja saat kondisi sistem dalam keadaan *state Wake* , pin yang diperlukan untuk dihubungkan yaitu pin **VCC** ditempatkan pada pin digital 8 dan pin **GND** pada Buzzer ke pin **GND** ATmega 328p. Skema rangkaian perancangan perangkat keras **buzzer** dapat dilihat pada Gambar 5.9 dan Tabel 5.5 merupakan perancangan yang digunakan pada rangkaian **buzzer**.



Gambar 5.9 Skematis Perancangan LED RGB

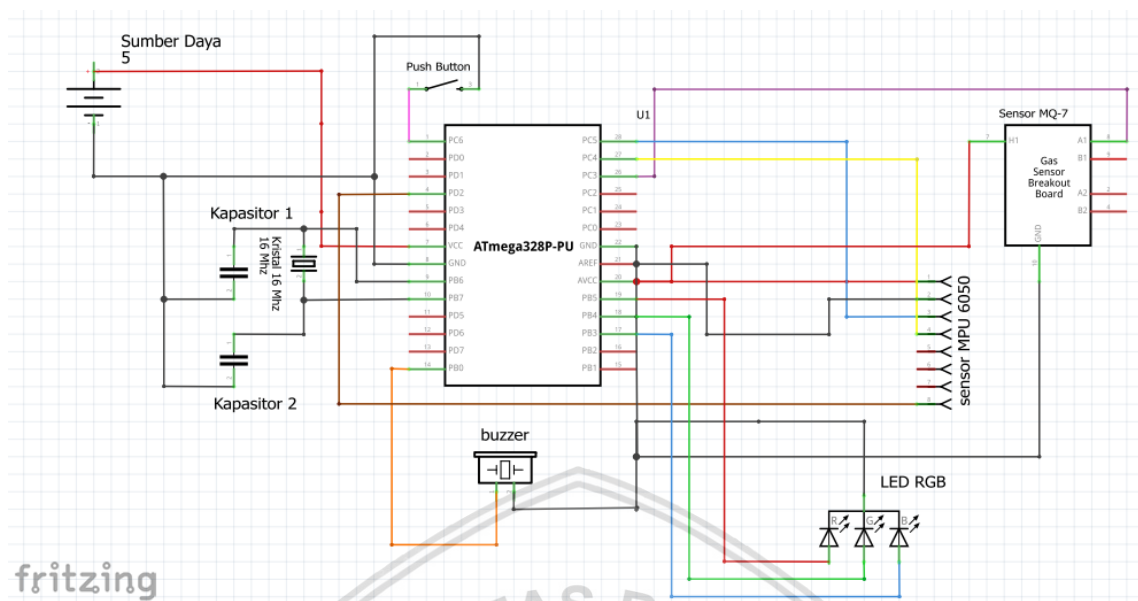
Tabel 5.5 Keterangan Pin Perancangan Pin Buzzer 5V

Buzzer	ATmega 328p	Keterangan
VCC	Pin 14	Pin digital 8
GND	Pin 8	GND

#### 5.1.4 Rangkaian Keseluruhan Sistem

Sistem yang akan dirancang menggunakan mikrokontroler ATmega328p sebagai titik kumpulnya pemrosesan data, ATmega328p dapat digunakan pada tegangan 5V dengan Crystal 16MHz. Terdapat dua modul sensor yang digunakan yaitu sensor mpu6050 dan sensor MQ-7. sensor MPU 6050 digunakan sebagai pendeteksi gerakan untuk memberikan *stimulus* pada mikrokontroler agar sistem dapat berpindah dari *state Power down Sleep* menjadi *state Wake*. Sensor MQ-7 membaca polutan yang ada pada kabin mobil, sensor MQ-7 membaca polutan di saat sistem pada kondisi *state Wake*, sensor MQ-7 terhubung dengan mikrokontroler pada pin *analog*.

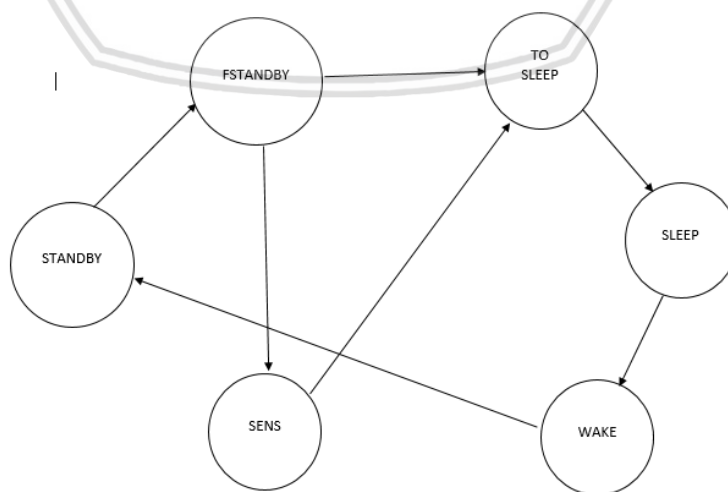
Untuk menampilkan data sistem mempunyai LED RGB untuk menampilkan informasi yang telah di olah oleh sistem. LED RGB terhubung dengan mikrokontroler menggunakan pin *digital*. Ada Buzzer sebagai *outputan* jika polutan melebihi batas toleransi yang ditetapkan oleh sistem, buzzer sendiri menggunakan pin *digital* sebagai *inputannya*. Untuk lebih jelasnya dapat dilihat pada skematis di Gambar 5.10 dibawah ini.



Gambar 5.10 Skematis Keseluruhan Alat

### 5.1.5 Perancangan Perangkat Lunak

Pada subbab ini merupakan bagian di mana perancangan perangkat lunak dari sistem yang akan dibuat dibahas. Pada perancangan perangkat lunak sistem akan dibahas mengenai metode *state machine* yang digunakan, pengaturan *power down sleep mode* dari mikrokontroler. Perangkat *monitoring* yang dirancang menggunakan *state machine* dengan beberapa *state* yang bertransisi ke *state* lainnya berdasarkan *stimulus* yang diberikan dan juga dari sistem itu sendiri dengan mengatur *interval* waktu tertentu. Berikut gambar *state diagram* dari *monitoring* alat pada gambar 5.11.



Gambar 5.11 Gambar *State Diagram* Sistem

Bagian-bagian *state Finite State Machine* yang digunakan sudah dirancang sedemikian rupa dan dinamai agar lebih mudah dimengerti. Untuk mengetahui penjelasan dari masing-masing *state* yang akan digunakan maka dapat dilihat pada tabel 5.6 dibawah ini.

**Tabel 5.6 Tabel Penjelasan Elemen *State Machine***

No.	State	Penjelasan
1.	STANDBY	Dalam mode <i>standby</i> merupakan mode awal berjalannya sistem dan belum ada aksi. (Kalibrasi Sensor)
2.	S_STANDBY	Dalam mode <i>standby</i> dilakukan pengecekan apakah sistem telah dilakukan pengecekan apakah sistem sudah pernah melewati <i>Sleep</i> atau belum.
3.	S_SENS	<i>Sensing</i> merupakan <i>state</i> dimana sensor dapat memulai pembacaan polutan menggunakan sensor MQ-7.
4.	S_TOSLEEP	<i>To Sleep</i> merupakan <i>state</i> transisi dari S_STANDBY menuju ke <i>Sleep</i> atau jeda sebelum <i>Sleep</i> dan menghitung <i>timer</i> .
5.	S_SLEEP	<i>Sleep</i> merupakan <i>state</i> dimana sistem berhenti bekerja dan fitur-fitur yang digunakan proses dimatikan sehingga dapat mereduksi arus yang digunakan Atmega 328p.
6.	S_WAKE	<i>Wake</i> merupakan <i>state</i> di mana sistem mulai diaktifkan kembali dan fitur yang telah dinonaktif akan nyala kembali untuk lanjut ke proses berikutnya.

Dalam *state machine* terdapat *next state* yaitu langkah yang akan dilalui selanjutnya dalam eksekusi *program* di dalamnya. Untuk mengetahui alur dari sebuah *State Machine* yang akan dijalankan pada sistem dapat dilihat pada tabel 5.7 di bawah:

**Tabel 5.7 Tabel *Next State Transition State Machine***

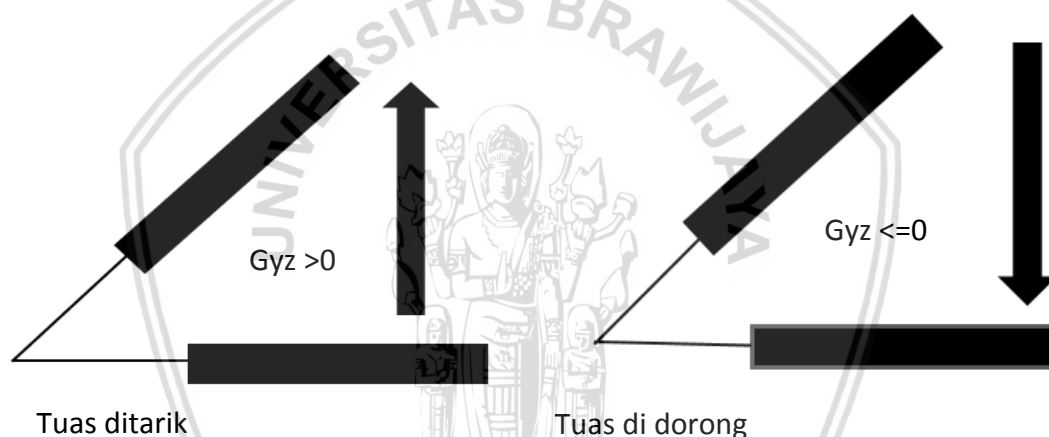
No.	State	Next State	Keterangan
1.	S_STANDBY	S_FSTANDBY	Mencatat waktu dan kalibrasi.
2.	S_FSTANDBY	S_SENS	Memeriksa kondisi <i>state</i> telah melewati <i>Sleep</i> atau belum, jika belum maka <i>state</i> berpindah ke sens.
3.	S_SENS	S_TOSLEEP	Saat kondisi <i>sensing</i> sedang berjalan namun <i>handbrake</i> diturunkan atau mobil berjalan maka <i>state</i> berpindah ke S_TOSLEEP.
3.	S_STANDBY	S_TOSLEEP	Saat kondisi status belum tidur sebelumnya dan telah melalui waktu 3 detik maka berpindah ke S_TOSLEEP.
4.	S_TOSLEEP	S_SLEEP	Kondisi status siap <i>Sleep</i> dikarenakan <i>timer</i> menuju sistem telah mencukupi.
5.	S_SLEEP	S_WAKE	Kondisi sistem saat <i>Sleep</i> dan terjadi gerakan yang dideteksi pin <i>Interrupt</i> maka sistem berubah ke <i>state</i> S_WAKE.
6.	S_WAKE	S_STANDBY	Saat kondisi <i>Wake</i> maka otomatis masuk ke <i>state</i> S_STANDBY

### 5.1.5.1 Perancangan *Sub System* Perangkat Lunak

Pada *subbab* ini merupakan bagian di mana perancangan perangkat lunak dari sistem yang akan dibuat dibahas. Pada perangkat lunak sistem akan dibahas mengenai metode *state machine* yang digunakan. Perangkat yang dirancang menggunakan *state machine* dengan beberapa *state* yang bertransisi ke *state* lainnya berdasarkan stimulus yang diberikan dan juga dari sistem itu sendiri dengan mengatur interval waktu tertentu.

#### a. Perancangan Perangkat Lunak *Motion detection*

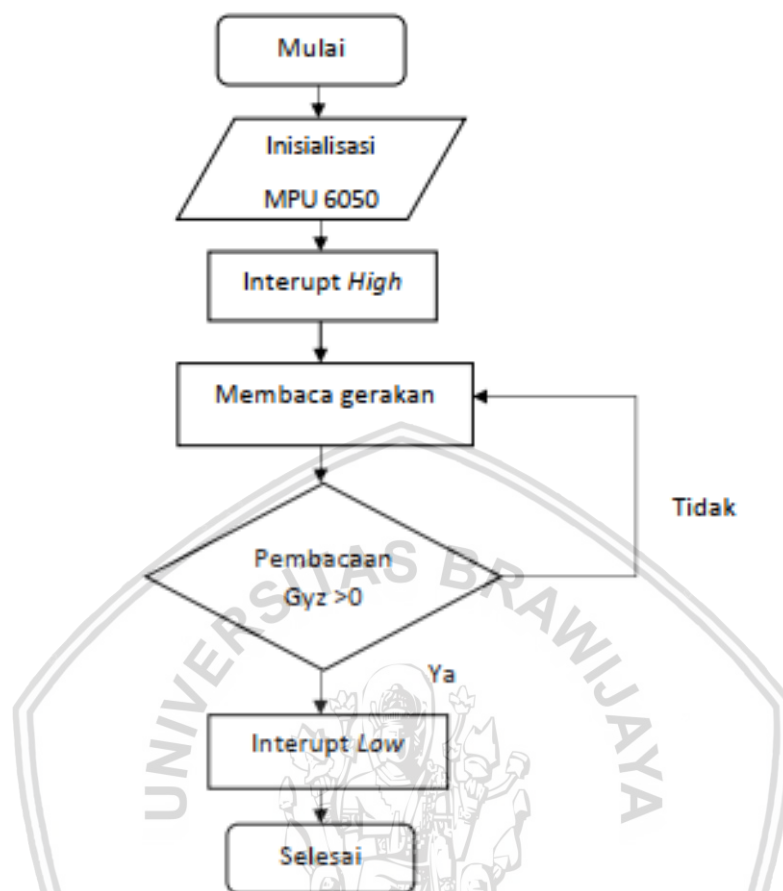
*Motion detection* dengan menggunakan MPU 6050 digunakan sebagai pemicu untuk membuat purwarupa sistem berpindah *state* dari *SLEEP* menjadi *WAKE*. Agar mudah di pahami maka simulasi pada handbrake dapat dilihat pada gambar 5.12 di bawah ini. Untuk melihat alur dari perancangan perangkat lunak maka dapat dilihat *flowchart* pada Gambar 5.13. Untuk melihat *flowchart* perancangan *motion detection* dapat dilihat pada gambar 5.14.



**Gambar 5.12 Simulasi *Motion Detection* pada *Handbrake* mobil**

Simulasi *Handbrake* dapat dilihat pada gambar di atas pada gambar 5.13 . Pada saat tuas *handbrake* berada pada posisi di tarik ke atas atau mobil sedang berhenti dan nilai *Gyz* menunjukkan nilai  $>0$  maka *interrupt* berlogika *LOW*. Sistem masuk dalam kondisi *Wake*. Jika posisi tuas *handbrake* berada pada posisi di dorong kebawah atau mobil sedang ingin di jalankan dan parameter *Gyz* menunjukkan  $<0$  maka *interrupt* berlogika *HIGH* maka sistem masuk dalam mode *Sleep*.

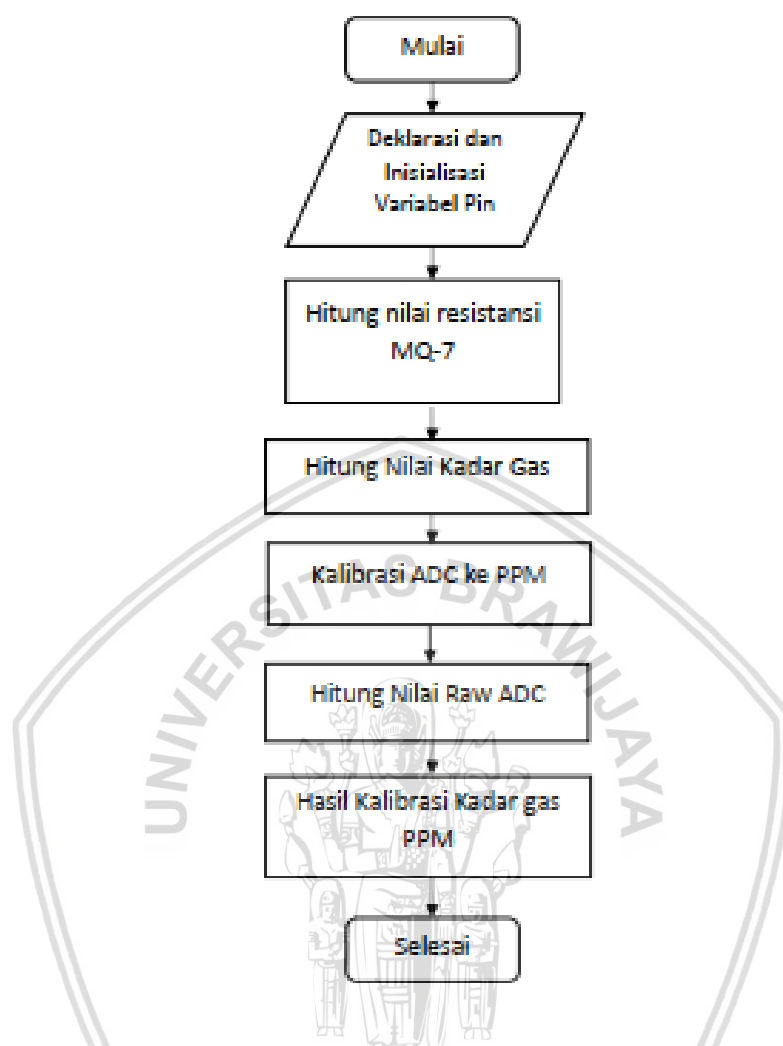




Gambar 5.13 *Flowchart Motion detection*

#### b. Perancangan Perangkat Lunak MQ-7 Sensing

Sensor MQ-7 yang digunakan untuk membaca perubahan polutan di dalam kabin mobil merupakan sensor analog. Pada penelitian yang dilakukan sensor MQ-7 menggunakan pin Analog 3 dari ATmega328p untuk dapat membaca perubahan polutan setiap 1 detik. Perangkat lunak MQ-7 akan dieksekusi ketika purwarupa sistem dalam posisi *state SENS*. *Flowchart* dari program sensor MQ-7 dapat dilihat pada Gambar 5.14 di bawah ini:

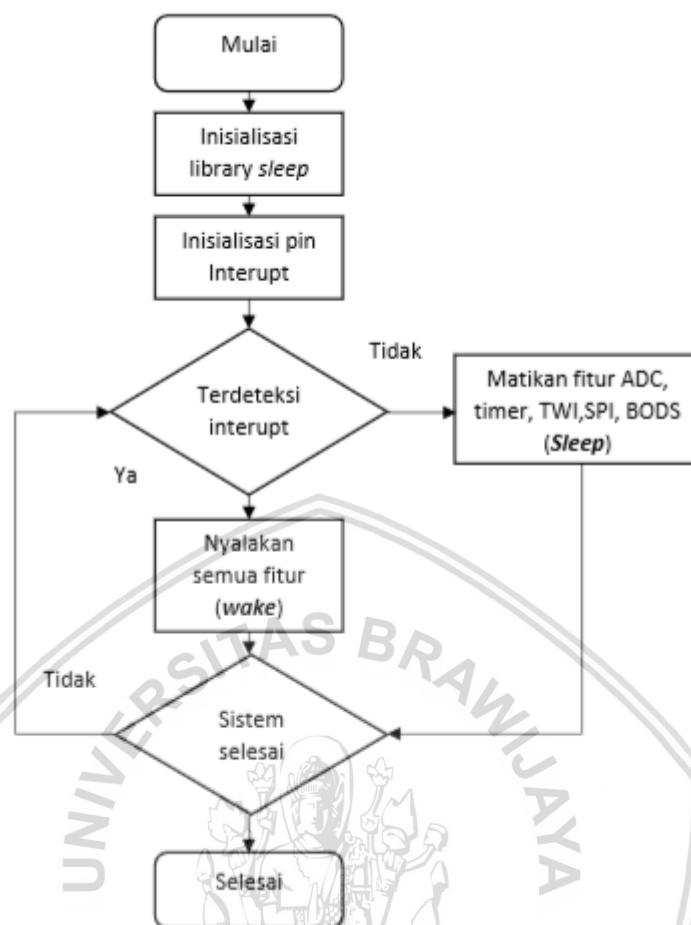


**Gambar 5.14 Flowchart Sensing Sensor MQ – 7**

Pertama dilakukan deklarasi dan inisialisasi variabel yang digunakan di Atmega 328p, Selanjutnya hitung nilai resistansi MQ 7, Setelah mendapat nilai resistansi maka lanjut ke proses perhitungan nilai kadar gas, saat kadar gas telah ditentukan maka dilakukan kalibrasi dari pin ADC ke satuan PPM dilanjutkan perhitungan nilai Raw ADC dan nilai PPM telah berhasil ditemukan dan ditentukan dan ditampilkan saat *state sens*.

### **c. Perancangan Perangkat Lunak (*low power*) *Power down Sleep***

Untuk menghemat konsumsi arus sistem akan berpindah ke posisi *Sleep* karena ketika dalam keadaan *Sleep* konsumsi arus menurun. Berikut ini adalah diagram alir dari perancangan perangkat lunak *Power down Sleep*. Berikut ini merupakan *flowchart* program untuk *Power down Sleep* dapat dilihat pada Gambar 5.15:



**Gambar 5.15 Flowchart Power down Sleep**

Sebelum mengeksekusi perintah *Sleep* akan ditentukan mode *Sleep* yang akan digunakan, pada penelitian ini akan digunakan mode *Power down*. Untuk lebih memangkas penggunaan arus maka fungsi seperti ADC, *Timer*, TWI/I2C, SPI, dan BODS akan dimatikan karena tidak digunakan ketika *Sleep*. Untuk membangunkan mikrokontroler dari *Sleep* maka diatur *interrupt* sebagai metode *Wake* yang akan digunakan. Ketika mikrokontroler mendeteksi adanya *interrupt* maka mikrokontroler akan *Wake* dan mengaktifkan fungsi yang telah dimatikan.

## 5.2 Implementasi Sistem

Pada subbab ini akan dibahas mengenai implementasi dari sistem yang telah dirancang. Hasil perancangan yang akan diimplementasikan adalah perangkat keras sistem, perangkat lunak pada sistem dan *portable device*.

### 5.2.1 Implementasi Perangkat Keras Sistem

Pada subbab ini akan dibahas mengenai implementasi dari hasil perancangan yang telah dibuat sebelumnya. Implementasi perangkat keras meliputi bagian bagian sebagai berikut.

#### a. Implementasi Perangkat Keras Catu Daya

Setelah perangkat keras catu daya melalui tahap perancangan maka rancangan yang telah dibuat akan diimplementasikan menjadi sebuah purwarupa. Berikut ini merupakan gambar dari hasil implementasi berdasarkan perancangan yang dibuat sebelumnya.

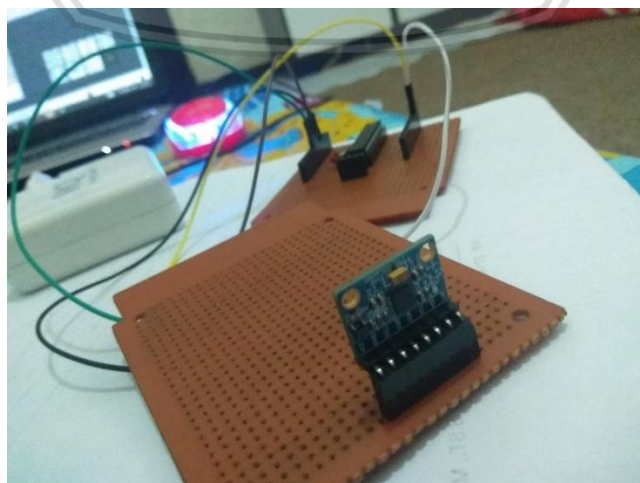
Catu daya dihubungkan dengan menggunakan connector head agar mudah untuk dicabut dan pasang. Kabel positif dari baterai dihubungkan ke pin VCC dan kabel negatif dihubungkan ke pin GND. Gambar Implementasi catu daya dapat dilihat pada Gambar 5.16 di bawah ini:



Gambar 5.16 Implementasi Catu Daya

#### b. Implementasi Perangkat Keras MPU 6050

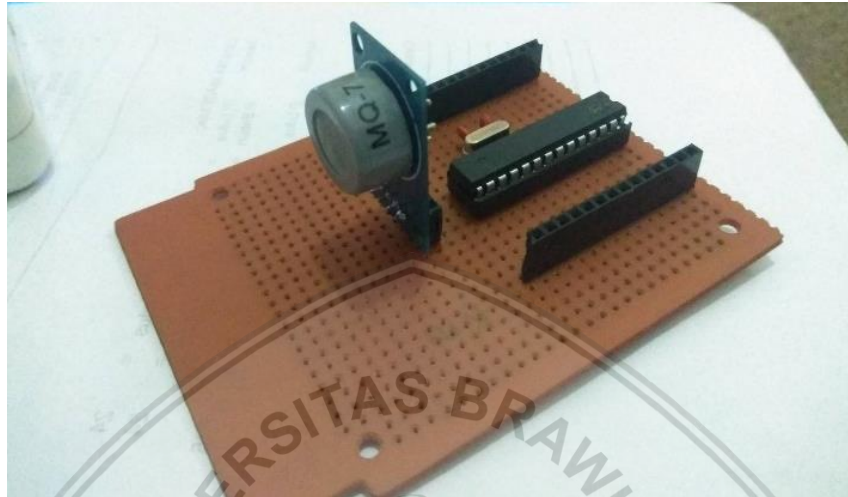
Setelah perangkat keras *Motion detection* melalui tahap perancangan maka hasil dari perancangan akan diimplementasikan menjadi sebuah purwarupa. Berikut ini merupakan gambar dari hasil implementasi perangkat keras *Motion detection*. Gambar Implementasi *Motion detection* dapat dilihat pada Gambar 5.17 di bawah ini:



Gambar 5.17 Implementasi perangkat keras MPU 6050

### c. Implementasi Perangkat Keras MQ-7

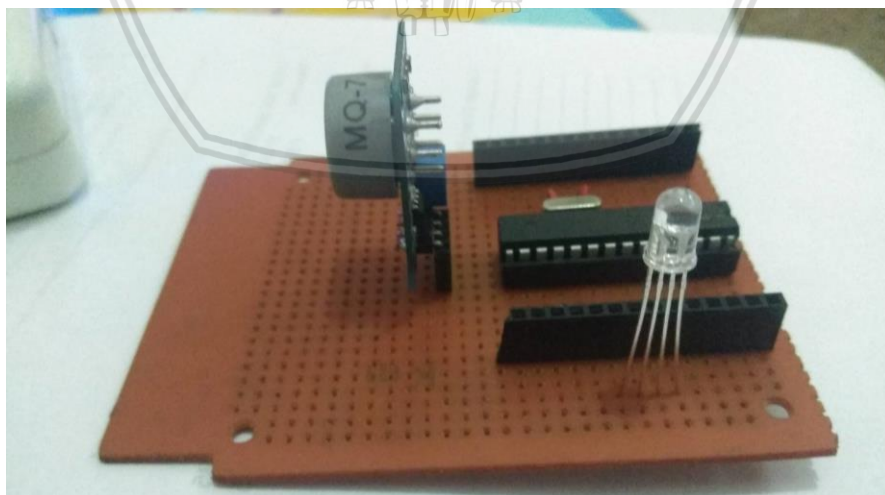
Setelah perangkat keras sensor MQ-7 melalui tahap perancangan, hasil dari perancangan akan diimplementasikan menjadi sebuah purwarupa. Berikut ini merupakan gambar dari hasil implementasi perangkat keras sensor MQ-7. Gambar Implementasi Sensor MQ-7 dapat dilihat pada Gambar 5.18 di bawah ini.



Gambar 5.18 Implementasi Perangkat Keras MQ-7

### d. Implementasi *LED RGB CC (common catode)*

Setelah perangkat keras *LED RGB CC* melalui tahap perancangan hasil dari perancangan akan diimplementasikan menjadi sebuah purwarupa. Berikut ini merupakan gambar dari hasil implementasi perangkat keras *LED RGB CC*. Gambar Implementasi *LED RGB* dapat dilihat pada Gambar 5.19 di bawah ini.

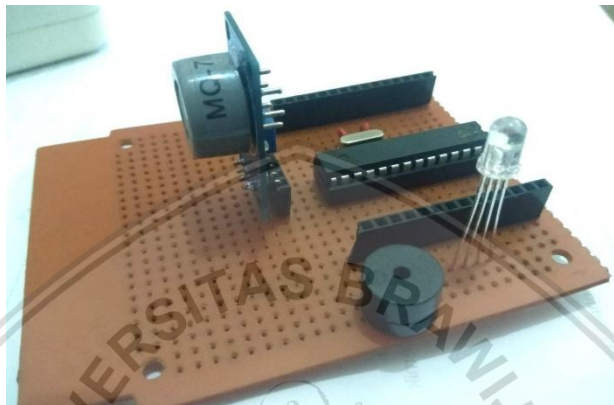


Gambar 5.19 Implementasi *LED RGB CC*



#### e. Implementasi Buzzer

Setelah perangkat keras *Buzzer* melalui tahap perancangan hasil dari perancangan akan diimplementasikan menjadi sebuah purwarupa. Berikut ini merupakan gambar dari hasil implementasi perangkat keras *Buzzer*. Gambar Implementasi *Buzzer* dapat dilihat pada gambar 5.20 di bawah ini.



Gambar 5.20 Implementasi Perangkat Keras *Buzzer*

#### f. Implementasi Penempatan Perangkat Keras Pada Mobil

Setelah melakukan proses perancangan penempatan letak posisi sistem, selanjutnya perancangan akan diimplementasikan menjadi sebuah purwarupa. Berikut ini merupakan gambar dari hasil implementasi perangkat keras keseluruhan sistem yang di tempatkan pada kabin mobil. Gambar Implementasi Perangkat Keras Pada mobil dapat dilihat pada gambar 5.21 di bawah ini.



Gambar 5.21 Implementasi Penempatan Perangkat Keras Pada Mobil



## 5.2.2 Implementasi Perangkat Lunak

Pada subbab ini akan dibahas mengenai implementasi dari hasil perancangan yang telah dibuat sebelumnya. Implementasi perangkat keras meliputi bagian - bagian sebagai berikut.

### a. Implementasi Perangkat Lunak *Finite state machine*

Hasil dari perancangan perangkat lunak *Finite state machine* Hasil akan diimplementasikan menjadi sebuah program yang telah dibuat menggunakan Arduino IDE. Program dari Perangkat lunak *Finite state machine* dapat dilihat pada Tabel 5.8:

**Tabel 5.8 Implementasi Perangkat Lunak *Finite state machine***

Program Finite State Machine	
1.	static int state = S_STANDBY;
2.	static unsigned long ts;
3.	switch (state) {
4.	case S_STANDBY:
5.	ts = millis();
6.	Serial.println("STANDBY STATE");
7.	state = S_FSTANDBY;
8.	break;
9.	case S_FSTANDBY:
10.	if (Sleepstat == false && millis() >= ts + 7000) {
11.	state = S_TOSLEEP;
12.	ts = millis();
13.	}
14.	else if (Sleepstat == true) {
15.	state = S_SENS;
15.	Sleepstat = false;
16.	ts = millis();
17.	} else
18.	return;
19.	break;
20.	break;
21.	case S_SENS:
22.	state = S_SENS;
23.	}
24.	if (acz <=0) {
25.	state = S_TOSLEEP;
26.	break;
27.	if (millis() > ts + 10000)
28.	state = S_STANDBY;
29.	delay(5000);
30.	break;
31.	case S_TOSLEEP:
32.	Serial.println("going to Sleep ");
33.	Serial.print("");
34.	if (millis() > ts + 2000) {
35.	state = S_SLEEP;
36.	Serial.println("_____SLEEP NOW!!!_____");
37.	Serial.print("");
38.	}
39.	delay(1000);

40.	break;
41.	case S_SLEEP:
42.	digitalWrite(BUZZER, LOW);
43.	digitalWrite(redPin, LOW); //if limit has been reached, LED
44.	turns on as status indicator
45.	digitalWrite(greenPin, LOW);
46.	digitalWrite(bluePin, LOW);
47.	state = S_WAKE;
48.	SleepNow();
49.	delay(1000);
50.	break;
51.	case S_WAKE:
52.	state = S_STANDBY;
53.	delay(1000);
54.	break;
55.	}
56.	}}

Dalam program di atas dapat dilihat jika sistem dibuat agar sesuai dengan yang di inginkan. Setiap *state* mempunyai *next state* dan *timer* masing - masing. Untuk awal mula sistem bekerja berada pada *state Standby* dan akan berpindah *state* jika syarat syarat yang di perlukan terpenuhi. Sistem akan berjalan terus menerus berpindah dari satu *state* ke *state* lainnya, sehingga alur program lebih rapi dan mempunyai tujuan. Untuk lebih rinci tentang *Finite state machine* dapat dilihat pada perancangan di Tabel 5.6 dan Tabel 5.7.

#### b. Implementasi Perangkat Lunak *Motion detection*

Hasil dari perancangan perangkat lunak *motion detection* akan diimplementasikan menjadi sebuah program yang telah dibuat menggunakan Arduino IDE. Potongan program dari Perangkat lunak *motion detection* dapat dilihat pada Tabel 5.9:

**Tabel 5.9 Potongan Program *Motion detection***

Potongan Program	
1.	Wire.beginTransmission(mpu);
2.	mpu.initialize();
3.	mpu.setInterruptMode(1); //mengatur mode interrupt
4.	mpu.setDHPFMode(1);
5.	mpu.setMotionDetectionThreshold(2);
6.	mpu.setMotionDetectionDuration(40);
	mpu.setMotionDetectionCounterDecrement(1);
7.	mpu.setIntMotionEnabled(1);
8.	Wire.endTransmission(true);
9.	attachInterrupt(0, WakeUpNow, LOW);

Untuk dapat menggunakan MPU 6050 diperlukan *library* untuk antarmuka komunikasi I2C (*Inter Integrated Circuit*) yaitu *Wire.h* dan juga *library* untuk MPU 6050 yaitu MPU 6050.h. Setelah *library* maka akan diinisialisasi alamat dari memori register dari MPU 6050 yaitu pada alamat 0x68. Untuk alamat tersebut tiap perangkat I2C memiliki standarisasi masing - masing agar tidak terjadi kesalahan pengiriman data. Setelah dilakukan inisialisasi alamat maka dilakukan

*instance* untuk fungsi dari *library* MPU 6050 dengan *variable* MPU. Untuk seterusnya dapat dilakukan komunikasi dengan MPU 6050 hanya dengan menggunakan *variable* MPU. Fungsi yang ada pada *library* juga dapat dipanggil dengan menggunakan *variable* MPU.

Pada potongan kode dari tabel 5.9 dilakukan inisialisasi untuk mpu6050 dan juga konfigurasi untuk *motion detection*. Akan diatur mode *interrupt* dari mpu6050 kemudian diatur filter *internal* dari MPU 6050 untuk mengurangi *noise* dengan frekuensi 5Hz, kemudian diatur *threshold* untuk batasan gerakan yang dideteksi hingga menghasilkan *interrupt*. Jika batasan dari *threshold* telah tercapai maka MPU 6050 akan menghasilkan logika *LOW* pada pin *Interrupt*.

### c. Implementasi Perangkat MQ-7

Implementasi dalam penggunaan perangkat lunak untuk mendeteksi gas pada kabin mobil menggunakan pin yang dijadikan sebagai acuan kode program dalam mengakses inisialisasi sistem. Pada potongan program di bawah ini merupakan inisialisasi MQ-7 yang sudah di tuning sesuai dengan kebutuhan sistem menentukan pin (A0) sebagai pin *inputan*. Tahanan beban di papan menggunakan 1 kilo ohm, resistansi sensor di udara bersih disesuaikan pada grafik *datasheet* MQ dan mendapatkan nilai (26,09), menentukan berapa banyak sampel dalam fase kalibrasi (50) dan lain sebagainya. Dalam menjalankan kinerja sistem nya, sensor gas diberikan variabel – variabel untuk dapat mengeksekusi kalibrasi sensor. Variabel – variabel tersebut dijadikan acuan dalam memberikan nilai hasil *outputan* berupa kadar gas pada CO. Berikut merupakan implementasi perangkat lunak potongan kode program yang digunakan dalam sensor gas ditunjukkan pada tabel 5.10.

**Tabel 5.10 Potongan Program MQ-7**

Potongan Program MQ-7			
1.	#define	MQ7PIN	(0)
2.	#define	RL_VALUE_MQ7	(1)
3.	#define	RO_CLEAN_AIR_FACTOR_MQ7	(26.09)
4.	#define	CALIBRATION_SAMPLE_TIMES	(50)
5.	#define	CALIBRATION_SAMPLE_INTERVAL	(50)
6.	#define	READ_SAMPLE_INTERVAL	(50)
7.	#define	READ_SAMPLE_TIMES	(5)

### d. Implementasi Perangkat Lunak (*low power*) *Power down Sleep*

Hasil dari perancangan perangkat lunak *Power down Sleep* akan diimplementasikan menjadi baris kode pada Arduino IDE. Hasil dari implementasi dapat dilihat pada Tabel 5.8 . Pada awalnya akan diinisialisasi *Sleep* dengan mode *Power down Sleep*. Fitur yang tidak digunakan seperti *ADC*, *timer*, *TWI*, *SPI*, *BODS* dan lainnya akan dimatikan, setelah itu akan diinisialisasi *interrupt* untuk *Wake* dari *Sleep*. Dengan pengaturan *Wake* ketika *interrupt* mendapati nilai logika *low*.

Setelah itu *Sleep* dieksekusi dan mikrokontroler menjadi *Sleep* hingga ada *interrupt* yang akan membuat mikrokontroler menjadi *Wake*. Ketika *Wake* maka *Sleep* akan dinonaktifkan dan fitur yang sebelumnya dimatikan akan dinyalakan kembali. Berikut merupakan implementasi perangkat lunak potongan kode program yang digunakan dalam sensor gas ditunjukkan pada Tabel 5.11.

**Tabel 5.11 Potongan Program *Power down Sleep***

Potongan Program <i>Power down Sleep</i>	
1.	<code>Sleep_enable();</code>
2.	<code>power_timer0_disable();</code>
3.	<code>power_timer1_disable();</code>
4.	<code>power_timer2_disable();</code>
5.	<code>power_twi_disable();</code>
6.	<code>power_adc_disable();</code>



## BAB 6 PENGUJIAN DAN ANALISIS

Pada bab 6 pengujian akan membahas mengenai pembahasan dari masing - masing fungsionalitas sistem yang terdiri dari pengujian fungsionalitas sensor MPU 6050, pengujian fungsionalitas sensor MQ - 7, pengujian fungsionalitas *LED RGB*, Pengujian *Buzzer*, pengujian arus ketika *Sleep*, dan pengujian keseluruhan sistem.

### 6.1. Pengujian Fungsionalitas sensor MPU 6050

Pengujian fungsionalitas sensor MPU 6050 merupakan pengujian yang dilakukan dengan menggunakan sensor MPU 6050 di mana akan di uji dengan melihat kondisi posisi dari sebuah *handbrake*. Hasil dari pengujian akan dianalisis.

#### 6.1.1 Tujuan

Untuk menguji sensor MPU 6050 yang menghasilkan *outputan* sudut dari pengguna ke sistem. Dari sini akan diketahui fungsionalitas sensor MPU 6050 dalam membaca perubahan posisi di *handbrake* Mobil.

#### 6.1.2 Prosedur Pengujian

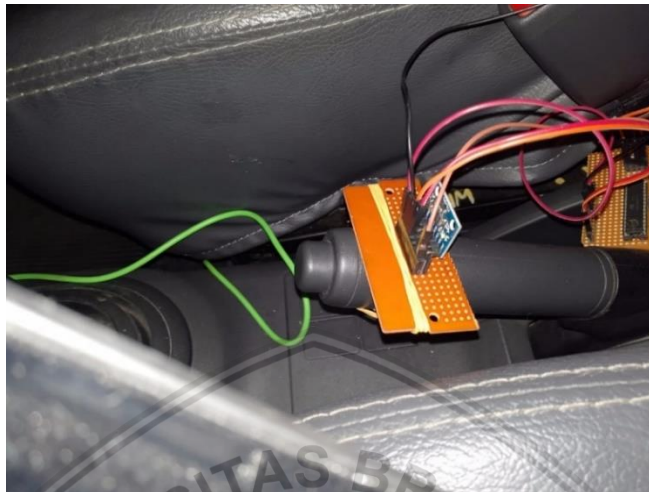
Agar dapat menguji sensor dalam memulai membaca nilai gas maka berikut alur yang dilakukan untuk mendapatkan pengambilan data sensor MPU 6050 sebagai berikut:

1. Persiapkan sensor *gyro* MPU 6050, mikrokontroller Atmega 328p dan beberapa *jumper*.
2. Sambungkan sensor MPU 6050 dengan jumper yang terhubung ke pin VCC, GND, Pin 28 untuk SCL, Pin 27 untuk SDA dan pin 4 untuk Interrupt
3. Buka software untuk mengunggah kode program yakni *IDE*.
4. Unggah *source code* kode program ke dalam *mikrokontroller* ATmega328p.
5. Pasang *case* sensor yang di ikatkan pada *Handbrake* mobil
6. Uji Tarik *handbrake* dan dorong *handbrake* untuk mengetahui nilai yang diinginkan
7. Amati dan catat setiap *inputan* sensor MPU 6050 dalam membaca perpindahan antara mobil berhenti dan berjalan
8. Setelah pengujian selesai dilakukan, ambil kesimpulan dari hasil pengujian kemudian dianalisis.

#### 6.1.3 Hasil dan Analisis

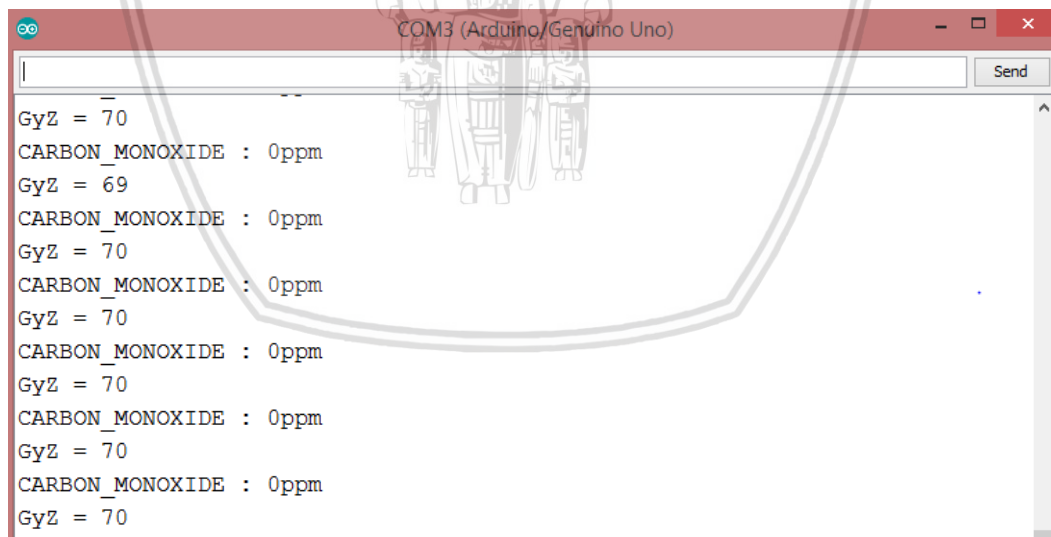
Usai dilakukan runtutan proses alur pengujian, langkah yang selanjutnya dapat menghasilkan pengujian sensor MPU 6050 dalam melakukan pembacaan

terhadap nilai Gyz yang diterapkan pada *Handbrake* mobil. Adapun hasil pengujian sensor *gyro* MPU 6050 dipaparkan pada Tabel 6.1. Untuk dokumentasi foto pengujian saat *handbrake* ditarik dapat dilihat pada gambar di bawah ini. Hasil dari *outputan* Serial Monitor dapat dilihat pada di bawah.




**Gambar 6.1 Pengujian MPU di *Handbrake* saat ditarik**

Gambar 6.1 merupakan pengujian sensor MPU 6050 ketika diterapkan dalam *handbrake* mobil secara langsung ketika ditarik sensor menunjukkan akan di *range* nilai raw data 70 yang dapat dilihat pada gambar 6.2 yang merupakan *outputan* dari serial monitor.



**Gambar 6.2 Hasil *outputan* dari Serial Monitor saat *Handbrake* di Tarik**





The screenshot shows the Arduino IDE serial monitor window titled "COM3 (Arduino/Genuino Uno)". The window has a search bar at the top and a "Send" button. The serial output displays the following text:

```
CARBON_MONOXIDE : 3ppm  
GyZ = 68  
CARBON_MONOXIDE : 5ppm  
GyZ = 71  
CARBON_MONOXIDE : 2ppm  
GyZ = 69  
CARBON_MONOXIDE : 2ppm  
GyZ = 68  
CARBON_MONOXIDE : 3ppm  
GyZ = 69  
CARBON_MONOXIDE : 4ppm  
GyZ = 1  
CARBON_MONOXIDE : 5ppm  
GyZ = -1  
going to sleep  
____SLEEP NOW!!!____
```

**Gambar 6.4 Hasil *outputan* dari Serial Monitor saat *Handbrake* didorong**

**Tabel 6.1 Tabel Hasil Pengujian Sensor MPU 6050**

Pengujian ke-	Kondisi	Hasil Gyz	Kondisi Sistem	Keterangan
1	<i>Handbrake</i> ditarik	70	<i>Wake</i>	Berhasil
2	<i>Handbrake</i> Turun	-1	<i>Sleep</i>	Berhasil
3	<i>Handbrake</i> ditarik	69	<i>Wake</i>	Berhasil
4	<i>Handbrake</i> Turun	-3	<i>Sleep</i>	Berhasil
5	<i>Handbrake</i> ditarik	71	<i>Wake</i>	Berhasil
6	<i>Handbrake</i> Turun	-1	<i>Sleep</i>	Berhasil
7	<i>Handbrake</i> ditarik	70	<i>Wake</i>	Berhasil

Pada Tabel 6.1 setelah melakukan percobaan sensor diuji pada *handbrake* mobil secara real dapat dilihat di serial monitor bahwa ketika *handbrake* ditarik nilai yang di hasilkan oleh sensor bernilai sekitar *range* 70 dan ketika *handbrake* diuji didorong yang menandakan mobil sedang akan berjalan mengeluarkan hasil sekitar *range negative*. Sehingga dapat dipastikan fungsional sensor bekerja sesuai yang diharapkan, karena memenuhi syarat sebagai parameter *interrupt* untuk memicu *wake* dan *sleep* pada ATmega 328p.

## 6.2 Pengujian Fungsionalitas sensor MQ-7

Pengujian fungsionalitas sensor MQ-7 merupakan bagian pengujian yang dilakukan dengan menggunakan sensor MQ-7 di mana sensor akan diuji sesuai kondisi lingkungan di mana mobil itu dinyalakan.

### 6.2.1 Tujuan

Untuk menguji sensor MQ-7 yang mendeteksi adanya polutan di dalam mobil. Dalam masa pengujian dilakukan menggunakan kondisionalitas ketika mobil ada di dalam ruangan dan di luar ruangan agar dapat diketahui perbedaannya.

### 6.2.2 Prosedur Pengujian

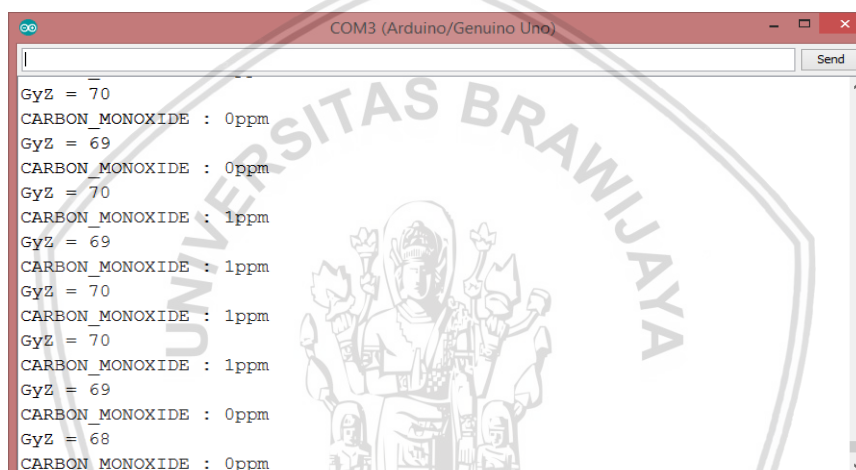
Agar dapat menguji sebuah sensor dalam membaca perubahan nilai gas maka berikut alur proses yang dilakukan untuk mendapatkan pengambilan data sensor gas MQ-7 sebagai berikut:

1. Persiapkan sensor gas MQ-7, mikrokontroler ATmega328p, dan beberapa *jumper*.
2. Sambungkan sensor MQ-7 dengan *jumper* yang terhubung ke pin VCC, GND dan Pin 23 pada mikrokontroler ATmega328p.
3. Bukalah *software* untuk mengunggah kode program yakni *IDE*.

4. Unggah *source code* kode program sensor dalam mikrokontroler ATmega328p.
5. Nyalakan kendaraan untuk mengambil data berupa nilai dari sensor gas.
6. Amati dan catat setiap *inputan* sensor MQ-7 dalam membaca nilai gas Karbon monoksida.
7. Setelah pengujian selesai dilakukan, ambil kesimpulan dari hasil pengujian kemudian dianalisis.

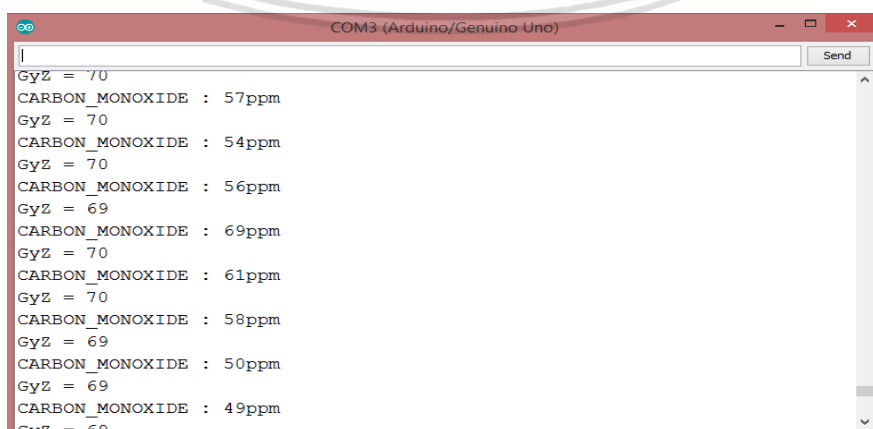
### 6.2.3 Hasil dan Analisis

Setelah melakukan semua proses alur pengujian, maka langkah selanjutnya mengamati hasil dari *output* yang di hasilkan dari sensor MQ- 7. Untuk hasil yang telah di amati dapat dilihat pada Gambar dan Tabel di bawah ini:



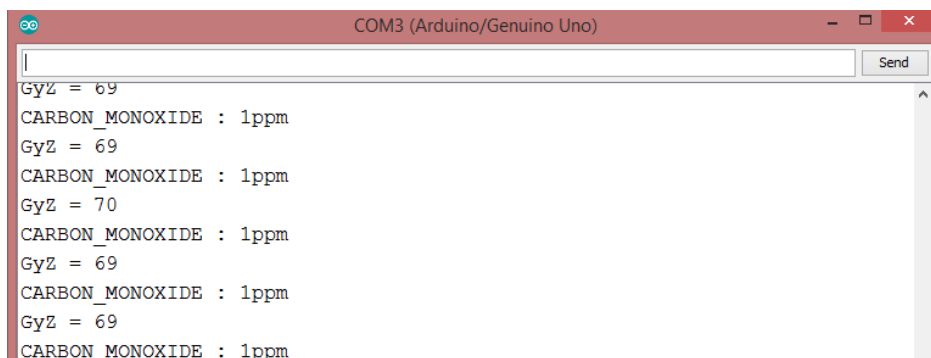
**Gambar 6.5 Hasil ppm ketika mobil tidak dinyalakan dan belum ada gas**

Gambar 6.5 merupakan hasil dari pengujian MQ 7 ketika di udara bebas yang bersih dari gas karbon monoksida, sehingga dapat dilihat pada layar serial monitor yang menampilkan ppm 0.



**Gambar 6.6 Hasil ppm ketika mobil sudah nyala dan terdeteksi gas**

Gambar 6.6 di atas merupakan hasil dari pengujian MQ-7 ketika di udara yang mengandung polutan karbon monoksida, sehingga dapat dilihat pada layer serial *monitor* yang menampilkan ppm 69.



**Gambar 6.7 Hasil ketika kaca jendela dibuka dan udara terurai**

Gambar 6.7 di atas merupakan hasil dari pengujian MQ – 7 ketika karbon monoksida mulai terurai di udara bersih sehingga ppm mulai berangsur menurun mendekati ppm 0 seperti awal saat di udara bersih.

**Tabel 6.2 Tabel Pengujian Sensor MQ-7 Pada Pembacaan PPM**

No	Kondisi	Pengujian Ke-	Hasil Rata-rata
1.	Ketika Mobil Tidak Dinyalakan dan Tidak Ada Gas	1	0 ppm
		2	0 ppm
		3	0 ppm
		4	0 ppm
		5	0 ppm
		6	0 ppm
		7	0 ppm
2.	Ketika Mobil Nyala dan Gas Masuk Kabin	1	66 ppm
		2	47 ppm
		3	45 ppm
		4	44 ppm
		5	43 ppm
		6	44 ppm
		7	44 pm
3.	Ketika Mobil nyala dan Kaca Dibuka	1	3 ppm
		2	2 ppm
		3	3 ppm
		4	2 ppm
		5	4 ppm
		6	3 ppm
		7	2 ppm

Pada Tabel 6.2 setelah mendapatkan hasil proses pengujian, mulanya sensor ketika di udara bersih sensor mengeluarkan data 0 ppm dikarenakan udara bersih dari karbon monoksida, saat kendaraan mulai dinyalakan sensor MQ -7 mendeteksi adanya perubahan data yang masuk ke dalam sensor, sensor mendeteksi sekitar 66 ppm menandakan adanya polutan yang berbahaya dan ketika polutan mulai terurai ketika kaca dibuka maka sensor mendeteksi sekitar 2 ppm yang mendekati udara bersih ketika awal belum dinyalakan mesin kendaraan.

### 6.3 Pengujian Notifikasi *LED RGB*

Pengujian Notifikasi *LED RGB* merupakan bagian pengujian dengan menggunakan *LED RGB* di mana lampu akan diuji ke akuratnya memberi informasi ke pengguna sesuai harapan sistem.

#### 6.3.1 Tujuan

Untuk menguji *LED RGB* yang dapat menghasilkan *outputan* warna yang sudah disesuaikan oleh keinginan pembuat program. Dari sini akan diuji apakah *outputan* warna LED sesuai dengan program yang dibuat.

#### 6.3.2 Prosedur Pengujian

1. Persiapkan *LED RGB*, Atmega 328p, dan beberapa *jumper*.
2. Sambungkan LED pada pin D11 untuk menyalakan warna merah, D12 untuk mengaktifkan warna hijau, D13 untuk mengaktifkan warna biru dan pin *Ground*.
3. Buka *software* untuk mengunggah kode program yakni *IDE*.
4. Unggah *source code* ke dalam mikrokontroler ATmega328p.
5. Beri *inputan* berupa polutan agar LED dapat bekerja.
6. Amati hasil warna dari *outputan* LED.
7. Setelah pengujian selesai dilakukan, ambil kesimpulan dari hasil pengujian kemudian dianalisis.

#### 6.3.3 Hasil dan Analisis

Setelah melakukan semua proses alur pengujian, maka langkah selanjutnya dapat menghasilkan pengujian sensor *LED RGB*. Dalam melakukan pemantauan *output* dari *LED RGB* dibagi dalam 3 kondisi ketika warna Merah, Biru dan Hijau Adapun hasil pengujian *LED RGB* dipaparkan pada Tabel 6.3.

**Tabel 6.3 Tabel Pengujian Kondisi *LED RGB***

No	Pengujian Ke -	PPM	Warna <i>Led</i>	Keterangan
1.	1	45	Merah	Berhasil

	2	46	Merah	Berhasil
	3	44	Merah	Berhasil
	4	44	Merah	Berhasil
	5	45	Merah	Berhasil
	6	43	Merah	Berhasil
	7	42	Merah	Berhasil
2.	1	22	Biru	Berhasil
	2	21	Biru	Berhasil
	3	23	Biru	Berhasil
	4	22	Biru	Berhasil
	5	22	Biru	Berhasil
	6	21	Biru	Berhasil
	7	23	Biru	Berhasil
3.	1	2	Hijau	Berhasil
	2	3	Hijau	Berhasil
	3	2	Hijau	Berhasil
	4	2	Hijau	Berhasil
	5	2	Hijau	Berhasil
	6	3	Hijau	Berhasil
	7	4	Hijau	Berhasil

Pada Tabel 6.2 Setelah mengamati dari pengujian dari LED RGB yang digunakan oleh sistem maka dapat dipastikan berhasil dikarenakan ketika sistem diberi *inputan* bernilai 45 ppm maka LED yang berwarna merah menyala, Ketika sistem diberi *inputan* bernilai 22 ppm maka LED yang berwarna Biru menyala dan ketika sistem diberi *inputan* bernilai 2 ppm maka LED yang berwarna hijau menyala.

#### 6.4 Pengujian Notifikasi *Buzzer*

Pengujian notifikasi *Buzzer* merupakan sebuah pengujian yang dilakukan untuk mengetahui akurasi alat dalam memberikan notifikasi. *Buzzer* akan diuji ketika sistem mendeteksi adanya polutan maka *buzzer* harus menyala agar bisa dianalisis ke akuratnya.



### 6.4.1 Tujuan

Untuk menguji *Buzzer* yang menghasilkan *outputan* suara ketika sistem mengisyaratkan adanya polutan yang berbahaya di dalam kabin. Dari sini akan diuji apakah *outputan* sesuai dengan yang diharapkan sistem.

### 6.4.2 Prosedur Pengujian

1. Siapkan *Buzzer*, Atmega 328p, dan 2 *jumper*.
2. Sambungkan kaki anoda pada pin 15 atau digital 9 dan katoda pada pin 22 atau Ground.
3. Buka software untuk mengunggah kode program yakni *IDE*.
4. Unggah *source code* kode ke dalam mikrokontroler ATmega328p.
5. Beri *inputan* berupa polutan yang tinggi agar *Buzzer* dapat bekerja
6. Dengarkan dan amati *output* dari *Buzzer*.
7. Setelah pengujian selesai dilakukan, ambil kesimpulan dari hasil pengujian kemudian dianalisis.

### 6.4.3 Hasil dan Analisis

Setelah melakukan semua proses alur pengujian, maka langkah selanjutnya dapat menghasilkan pengujian *output* buzzer. Dalam melakukan pemantauan *output* dari buzzer dibagi dalam 2 kondisi ketika ppm tinggi maka *buzzer* menyala terus menerus dan logika di pemrograman *high*. Ketika kondisi ppm rendah menyentuh batas parameter yang diinginkan maka logika pemrograman berubah menjadi *low*. Adapun hasil pengujian dapat dilihat pada tabel 6.4.

**Tabel 6.4 Tabel Pengujian Output Buzzer**

No.	Pengujian ke-	PPM	Output Buzzer	Keterangan
1.	1	46	<i>High</i>	Berhasil
	2	43	<i>High</i>	Berhasil
	3	45	<i>High</i>	Berhasil
	4	45	<i>High</i>	Berhasil
	5	43	<i>High</i>	Berhasil
	6	42	<i>High</i>	Berhasil
	7	44	<i>High</i>	Berhasil
2.	1	0	<i>Low</i>	Berhasil
	2	2	<i>Low</i>	Berhasil
	3	2	<i>Low</i>	Berhasil
	4	3	<i>Low</i>	Berhasil
	5	4	<i>Low</i>	Berhasil

	6	3	Low	Berhasil
	7	2	Low	Berhasil

Pada Tabel 6.4 setelah mendapatkan hasil proses pengujian, maka dapat menghasilkan *outputan High* ketika ppm yang masuk di olahan sistem tinggi. Untuk *outputan* mati harus dalam kondisi ppm tidak lebih dari 10 ppm. Untuk mendapat hasil pengujian harus berkaitan dengan sensor MQ-7 yang bertugas sebagai *inputannya*.

## 6.5 Pengujian Arus Ketika Sistem *Wake* dan *Sleep*

Untuk mengetahui konsumsi arus dari purwarupa sistem, pengukuran dilakukan ketika dalam keadaan *Sleep* konsumsi arus dibaca dengan menggunakan multimeter kemudian dibandingkan dengan konsumsi arus ketika sistem dalam keadaan *Wake*.

### 6.5.1 Tujuan

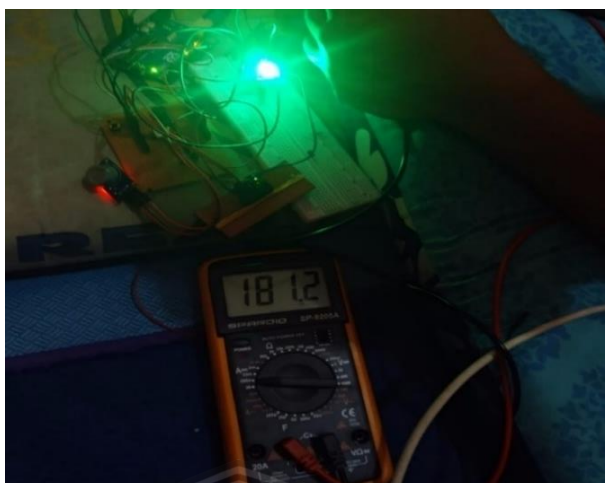
Tujuan dari pengujian Arus ketika *lowpower Sleep* dan *Wake* adalah untuk mengetahui besar arus yang digunakan dalam oleh sistem ketika dalam keadaan *Sleep* dan besar arus ketika sistem dalam keadaan *Wake*. Satuan yang digunakan untuk arus yang digunakan adalah mili *ampere* atau disingkat sebagai mA.

### 6.5.2 Prosedur Pengujian

1. Siapkan mikrokontroler ATmega328p dan *multitester*.
2. Sambungkan kaki positif *multitester* pada kaki GND ATmega328p dan kaki *negative multitester* pada kaki GND catu daya baterai.
3. Buka *software* untuk mengunggah kode *program* yakni *IDE*.
4. Unggah *source code* kode ke dalam mikrokontroler ATmega328p.
5. Amati mikrokontroller ATmega328p saat masuk mode *Sleep* dan *Wake*.
6. Amati setiap perubahan nilai pada *multitester* ketika memasuki kondisi *Sleep* dan *saat* kondisi *Wake*.
7. Setelah pengujian selesai dilakukan, ambil kesimpulan dari hasil pengujian kemudian dianalisis.

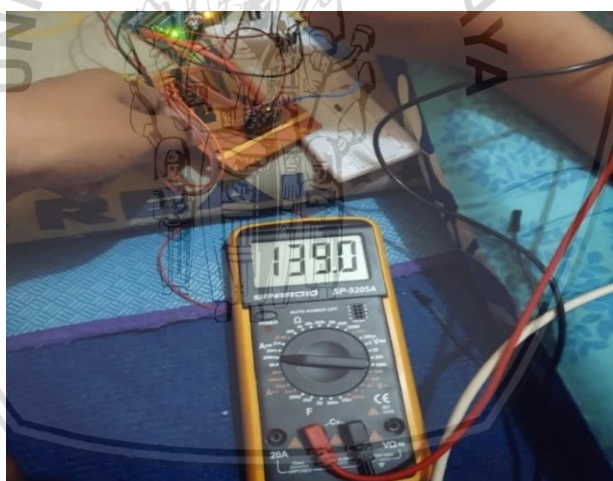
### 6.5.3 Hasil dan Analisis

Pengujian terhadap perbandingan konsumsi arus dalam melakukan pemrosesan mekanisme *low power Sleep mode* dan *Wake* menggunakan *external interrupt*. Adapun hasil pengujian terhadap pembacaan konsumsi arus pada kondisi *Wake* dipaparkan pada Tabel 6.5.



**Gambar 6.8 Pengukuran arus ketika sistem kondisi *Wake***

Pada gambar 6.8 merupakan pengukuran arus sistem yang mengalir pada saat kondisi *wake* sehingga dapat dilihat hasilnya melalui *multimeter* di atas, setelah mendapat nilai pengujian ketika *wake* maka pengujian dilanjutkan pada saat sistem *sleep*.



**Gambar 6.9 Pengukuran arus ketika sistem kondisi *Sleep***

Pada gambar 6.9 merupakan hasil dari pengujian arus sistem yang mengalir pada saat sistem kondisi *sleep*. Terdapat perbedaan nilai dari pengujian ketika kondisi *wake* dan *sleep* karena ketika *sleep* fitur pada ATmega yang tidak dibutuhkan telah dinonaktifkan.

**Tabel 6.5 Tabel Perbandingan Arus ketika *Sleep* dan *Wake***

Pengujian ke-	Arus Kondisi <i>Sleep</i>	Arus Kondisi <i>Wake</i>	Selisih Arus
1	139,0 mA	181,1 mA	42,1 mA
2	138,9 mA	181,2 mA	42,3 mA

3	139,0 mA	181,2 mA	42,2 mA
4	138,9 mA	181,2 mA	42,3 mA
5	138,9 mA	181,1 mA	42,2 mA
6	139,0 mA	181,1 mA	42,2 mA
7	138,9 mA	181,2 mA	42,3 mA
8	138,9 mA	181,2 mA	42,3 mA
Rata-Rata	139,2 mA	181,2 mA	42 mA

Pada Tabel 6.5 setelah mendapatkan hasil proses pengujian, maka dapat menghasilkan nilai dari pengujian konsumsi perbandingan konsumsi arus ketika kondisi *Sleep* dan *Wake*. Mikrokontroler ATmega328p dapat melakukan mekanisme *Sleep* apabila sistem tidak mendeteksi adanya nilai gas yang didapat dari pembacaan sensor gas. Kondisi *Wake* berjalan apabila sistem bermula pada posisi *Sleep* dan dibangunkan melalui sensor MPU 6050 menggunakan *external interrupt*. Arus yang berjalan pada kondisi *Wake* menunjukkan nilai rata – rata 181,2 mA dan arus yang berjalan pada kondisi *Sleep* menunjukkan nilai rata – rata 139,2 mA. Pengurangan jumlah arus yang terjadi yakni sebanyak 42 mA. Ini menunjukkan bahwa sistem berhasil menurunkan tingkat konsumsi daya dengan memanfaatkan fitur *sleep power down* yang mematikan beberapa fungsionalitas ATmega328P itu sendiri. Dengan begitu, pemanfaatan sumber daya energi *Wake* dapat diimplementasikan dan dapat berjalan sesuai dengan prinsip kerjanya untuk digunakan.

## 6.6 Pengujian fungsional keseluruhan sistem

Pengujian pada fungsional sistem dalam mengeksekusi sistem dilakukan dengan memasang dua buah sensor yakni sensor gas MQ-7 pada pin *analog* mikrokontroler dan sensor *gyro* MPU 6050 pada pin analog juga di pin SDA dan SCL. Untuk notifikasi sistem memasang *LED RGB* dan *buzzer* pada pin *digital* mikrokontroler. Pengujian fungsionalitas meliputi secara keseluruhan sistem melakukan kinerja prinsip kerjanya dan mengetahui keberhasilan sistem dalam menjalankan tiap – tiap kode program.

### 6.6.1 Tujuan

Untuk menguji sistem fungsional secara menyeluruh dalam membaca data dari sensor berupa nilai *analog* maupun nilai *digital*, penggunaan mekanisme *low power*, serta dapat memunculkan kondisi sistem dalam keadaan *Sleep* maupun *Wake*.

### 6.6.2 Prosedur Pengujian

Untuk dapat menguji fungsional secara menyeluruh maka berikut alur yang dilakukan untuk mendapatkan pengambilan data sensor suara FC-4 sebagai berikut:

1. Siapkan sensor gas MQ-7, sensor *gyro* MPU 6050, mikrokontroler ATmega328p, *LED RGB*, *buzzer*, dan beberapa *jumper*.
2. Sambungkan sensor gas MQ-7, Sensor *gyro* MPU 6050, *LED RGB*, *buzzer*, pada masing – masing pin mikrokontroler ATmega328p.
3. Buka *software* untuk mengunggah kode program yakni *IDE*.
4. Unggah *source code* kode program ke dalam mikrokontroler ATmega328p.
5. Pastikan Sensor *Gyro* sudah terpasang pada *Handbrake* mobil karena sebagai parameter *Sleep* dan *Wake* sistem.
6. Nyalakan Kendaraan untuk mengambil data berupa nilai ppm sensor gas.
7. Amati setiap *input* sensor gas dalam membaca nilai kadar gas karbon monoksida.
8. Amati *LED RGB* apakah sesuai dengan yang di inginkan peneliti.
9. Amati *Buzzer* apakah sesuai dengan yang di harapkan peneliti.
10. Setelah pengujian selesai dilakukan, ambil kesimpulan dari hasil pengujian kemudian dianalisis.

### 6.6.3 Hasil dan Analisis

Setelah melakukan semua proses alur pengujian, maka langkah selanjutnya dapat menghasilkan pengujian fungsional sistem secara menyeluruh dalam melakukan pembacaan terhadap nilai gas dan perpindahan posisi *handbrake* yang ditarik dan didorong serta mekanisme *sleep mode* dan *wake*. Adapun hasil pengujian fungsional sistem secara menyeluruh dipaparkan pada Tabel 6.6.

**Tabel 6.6 Hasil Pengujian Keseluruhan sistem**

Pengujian ke -	Pembacaan sensor MPU 6050	Pembacaan sensor MQ-7	State FSM	Status Wake	Status Sleep	LED RGB	Buzzer
1	70	45	<i>Sens</i>	✓	-	merah	<i>high</i>
2	-1	-	<i>Sleep</i>	-	✓	-	-
3	70	0	<i>Sens</i>	✓	-	hijau	<i>low</i>



4	-1	-	<i>Sleep</i>	-	✓	-	-
5	69	50	<i>Sens</i>	✓	-	merah	<i>high</i>
6	-2	-	<i>Sleep</i>	-	✓	-	-
7	71	2	<i>Sens</i>	✓	-	hijau	<i>low</i>
8	-1	-	<i>Sleep</i>	-	✓	-	-
9	-2	15	<i>Sens</i>	✓	-	biru	<i>low</i>
10	70	-	<i>Sleep</i>	-	-	-	-

Pada Tabel 6.6 telah dilaksanakan pengujian yang menghasilkan pengujian sukses, Sensor MQ-7 dapat menangkap nilai polutan yang ada pada kendaraan, Sensor MPU 6050 dapat mendeteksi kendaraan sedang berjalan atau berhenti sehingga mobil ketika berjalan berhasil *Sleep* agar sumber daya tidak terbuang sia - sia. Notifikasi berupa *LED RGB* sesuai dengan yang diharapkan jika berbahaya menyala merah, jika sedang berwarna biru dan berwarna hijau jika kondisi aman, Serta *Buzzer* berhasil memberikan peringatan ketika sistem mendeteksi polutan banyak dan berbahaya. Mekanisme *low power* pada keadaan *sleep* akan dijalankan ketika tidak mendeteksi adanya perubahan posisi *handbrake*, namun ketika keadaan sistem *wake* sensor MQ-7 masih mendapatkan nilai gas karbon monoksida tetapi tuas *handbrake* didorong yang menandakan mobil sedang berjalan maka sistem harus *sleep* sesuai dengan program yang dituliskan.



## BAB 7 KESIMPULAN DAN SARAN

Pada bab penutup yakni membahas mengenai kesimpulan dari proses penelitian yang dilakukan tentang “Pengembangan *Low Power System Monitoring* Karbonmonoksida (CO) Menggunakan Metode *State Machine*” serta saran – saran yang membangun dalam penelitian untuk dapat dikembangkan pada penelitian lebih lanjut.

### 7.1 Kesimpulan

Kesimpulan pada penelitian yang dikerjakan berdasarkan bentuk perancangan dan hasil implementasi, pengujian perancangan dan analisis pada sistem maka menghasilkan beberapa kesimpulan yang dapat diambil dalam beberapa pemaparan. Adapun pemaparan tersebut dapat dijabarkan sebagai berikut:

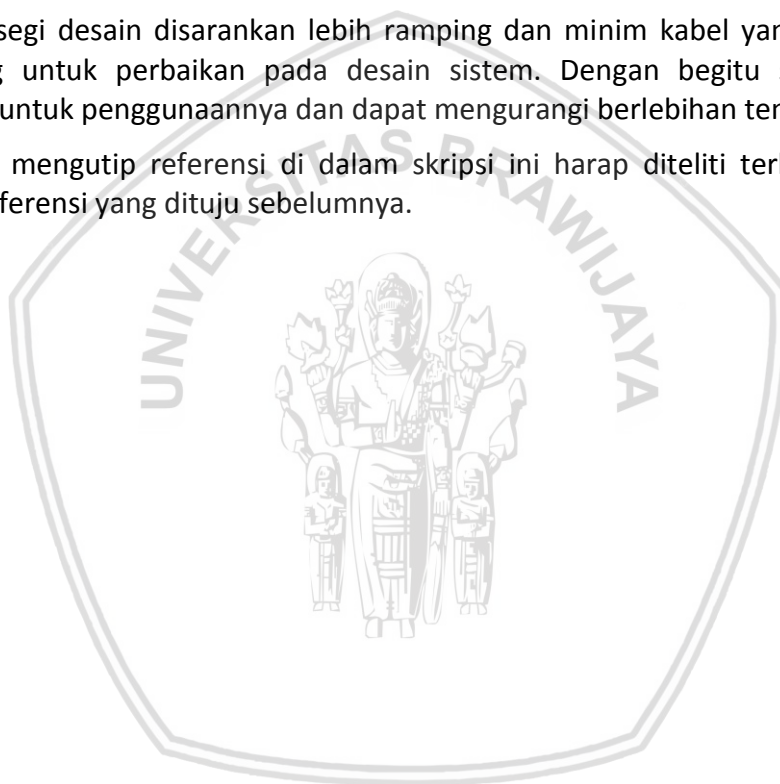
1. Dalam melakukan proses perancangan sistem, penelitian ini menggunakan fitur yang disediakan oleh Atmega 328p yaitu *Power Down Sleep*. Sistem menonaktifkan Atmega 328p dan untuk mengaktifkan *sleep mode* atau *wake mode* pada Atmega 328p akan menggunakan sensor *gyro* MPU 6050 yang akan digunakan sebagai alat pemicu untuk bisa masuk mode *sleep* dan *wake*. Parameter sensor MPU 6050 didapat berdasarkan pergerakan handbrake mobil ketika di tarik hingga batas *threshold* terpenuhi maka sistem masuk ke mode *sleep* dan ketika *handbrake* di dorong kebawah hingga batas *threshold* terpenuhi maka sistem masuk ke mode *wake*. Untuk Perancangan sistem yang berdasarkan sensor MQ – 7 yang sensor harus diletakkan pada kabin mobil yang berfungsi sebagai *inputan* kedua pada sistem dan parameternya berdasarkan berapa Ppm polutan yang masuk ke dalam mobil tersebut.
2. Untuk metode *state machine* yang di gunakan pada penelitian kali ini sudah berjalan sesuai yang diharapkan oleh peneliti. Ketika Kondisi baru dinyalakan sistem masuk dalam *state standby* untuk mengolah kalibrasi, Ketika *state* masuk ke dalam *fase sensing*, sistem dapat mendeteksi adanya polutan di dalam mobil, ketika inputan dari sensor MPU 6050 berlogika *high* maka sistem dapat berpindah *state* ke posisi *to sleep* dan menuju ke *sleep* hingga mendapatkan logika *low* agar dapat berpindah posisi ke *wake*.
3. Performa pengukuran kepada pengujian yang telah dilakukan mendapat hasil yang sesuai sebagai acuan bekerjanya sistem. Sensor gas yang dapat mendeteksi gas CO, Sensor *gyro* yang dapat menentukan kondisi mobil sedang berjalan atau berhenti. Sistem dapat menentukan ketika berhenti harus *Wake* dan ketika jalan harus *Sleep*. Sistem berhasil mereduksi daya sekitar 42 mA karena saat kondisi *Wake* di dapat rata-rata 139,2 mA dan ketika *Sleep* mendapat rata rata 181,2 mA.

### 7.2 Saran

Saran pada penelitian yang dikerjakan berdasarkan pemaparan kesimpulan yang sudah disebutkan agar dapat dikembangkan lebih lanjut berdasarkan

penelitian yang dilakukan maka dipaparkan beberapa saran. Adapun saran tersebut dapat dijabarkan sebagai berikut:

1. Untuk pengembangan lebih lanjut bisa menggunakan sensor *accelero* sebagai parameter mobil sedang berjalan atau berhenti. Ditambahkan *actuator* berupa *Motor Servo* agar ketika pengguna tidak menghiraukan maka sistem otomatis menggerakkan *servo* untuk membuka kaca mobil.
2. Dapat ditambahkan *wireless* untuk komunikasi antara sensor *Gyro* dengan *Atmega* yang letaknya tidak dalam satu tempat. Karena dalam penelitian ini menggunakan kabel jumper untuk jembatan komunikasi datanya. Sehingga rawan terjadinya putus kabel saat sedang digunakan.
3. Dalam segi desain disarankan lebih ramping dan minim kabel yang panjang – panjang untuk perbaikan pada desain sistem. Dengan begitu sistem lebih mudah untuk penggunaannya dan dapat mengurangi berlebihan tempat.
4. Apabila mengutip referensi di dalam skripsi ini harap diteliti terlebih dahulu pada referensi yang dituju sebelumnya.



## DAFTAR PUSTAKA

- Arduino, (2014)., Dari : <https://www.arduino.cc/>. [Diakses pada 25 Februari 2018]
- Components 101, n.d. x Dari : <https://components101.com/buzzer-pinout-working-datasheet>. [Diakses pada 15 Mei 2018]
- Device, M., n.d. *Modern Device.*, Dari : <https://moderndevice.com/product/rgb-leds-common-anode/> .[Dakses pada 17 Mei 2018]
- Ihsannurahim, 2018. *Implementasi Low Power Wearable Device Sebagai Heart Rate Monitor Dengan Metode State Machine*. [Diperoleh pada 25 februari 2018]
- Kuswara, 2006. Jurnal Teknik Lingkungan 3. *Inventori Emisi Polutan CO, Nox, HC dan SPM*, pp. 215-224. [Diperoleh pada 7 Mei 2018]
- Liputan6, 2018. *Pria Tewas di Dalam Mobil, Diduga Keracunan Mesin Pendingin*, Jakarta: Rio Audhitama Sihombing. [Diakses pada 8 Februari 2018]
- Munawar, A., 1999. *Traffic Accident Database Management System in Indonesia*. [Diperoleh pada 26 Februari 2018]
- Oracle, n.d. *Oracle Technology Network.*, Dari : <http://www.oracle.com/technetwork/systems/fsm-156381.html>. [Diakses pada 19 Mei 2018]
- Ortiz, C. E., 2004. *Managing the MIDlet Life-Cycle with a Finite State Machine.*, Dari : <http://www.oracle.com/technetwork/systems/fsm-156381.html>. [Diakses pada 27 Februari 2018]
- Purwanto, F., 2016. *Implementasi Sistem Monitoring Gas Berbahaya Pada Mobil Berbasis Logika Fuzzy Menggunakan Mikrokontroller*. [Diperoleh pada 23 Februari 2018]
- Rich, E., 2009. Automata, computability, and complexity. In: P. Education, ed. United State America: s.n. [Diakses pada 26 Februari 2018]
- Soedomo, M., 2003. In: *Kumpulan Karya Ilmiah Pencemaran Udara*. BANDUNG: ITB PRESS. [Diperoleh pada 28 Juni 2018]
- Wardhana, A. W., 2004. J&J Learning Yogyakarta. *Dampak Pencemaran Lingkungan*. [Diperoleh pada 28 Februari 2018]
- Yulianti, at aL, 2013. *Analisis Konsentrasi gas karbonmonoksida (CO) pada ruas jalan gajah mada pontianak* . [Diperoleh pada 25 Februari 2018]